5-2013

# The Dynamic Analysis of a Composite Overwrapped Gun Barrel with Constrained Viscoelastic Damping Layers Using the Modal Strain Energy Method

Braydon Day Hall
*Utah State University*

THE DYNAMIC ANALYSIS OF A COMPOSITE OVERWRAPPED GUN

BARREL WITH CONSTRAINED VISCOELASTIC

DAMPING LAYERS USING THE MODAL

STRAIN ENERGY METHOD

by

Braydon Day Hall

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

| | |
|---|---|
| Dr. Thomas Fronk<br>Major Professor | Dr. Steve Folkman<br>Committee Member |
| Dr. Leijun Li<br>Committee Member | Dr. Mark R. McLellan<br>Vice President for Research and<br>Dean of the School of Graduate Studies |

UTAH STATE UNIVERSITY
Logan, Utah

2013

# ABSTRACT

The Dynamic Analysis of a Composite Overwrapped Gun Barrel with

Constrained Viscoelastic Damping Layers Using the

Modal Strain Energy Method

by

Braydon Day Hall, Master of Science

Utah State University, 2013

Major Professor: Dr. Thomas H. Fronk
Department: Mechanical and Aerospace Engineering

The effects of a composite overwrapped gun barrel with viscoelastic damping layers are investigated. Interlaminar stresses and constrained layer damping effects are described. The Modal Strain Energy method is developed for measuring the extent to which the barrel is damped. The equations of motion used in the finite element analysis are derived. The transient solution process is outlined. Decisions for selected parameters are discussed. The results of the finite element analyses are presented using the program written in FORTRAN. The static solution is solved with a constant internal pressure resulting in a calculated loss factor from the Modal Strain Energy Method. The transient solution is solved using the Newmark-Beta method and a variable internal pressure.

The analyses conclude that strategically placed viscoelastic layers dissipate strain energy more effectively than a thick single viscoelastic layer. The optimal angle for maximizing the coefficient of mutual influence in a composite cylinder is not necessarily the optimal angle when viscoelastic layers are introduced between layers.

(200 pages)

# PUBLIC ABSTRACT

## The Dynamic Analysis of a Composite Overwrapped Gun Barrel with Constrained Viscoelastic Damping Layers Using the Modal Strain Energy Method

By using composite materials and rubber overwrapped onto a steel liner, a new design for a gun barrel is investigated. Multiple configurations are analyzed and compared using the Finite Element Method and analytical solutions. By alternating layers of composite material and rubber over the steel liner, different reactions in the barrel are obtained when subjected to identical internal pressures from a projectile being fired. The ideal reaction desired from creating this new gun barrel is to dissipate energy more effectively than gun barrels made entirely of steel or overwrapped with just a composite material. The analyses show that multiple thin layers of rubber are more effective than fewer, thicker layers. It is also demonstrated that the direction in which the composite material is overwrapped in between each rubber layer has an effect on dissipating energy.

Braydon Day Hall

For Grandpa Ozzie, the late KI6ZK,
It’s no muzzle loader but it could be one hell of gun!

# ACKNOWLEDGMENTS

To my family, friends, and colleagues who have supported and encouraged me over the last two years, thank you. I would like to thank all of the professors that I have had the privilege of learning from as a graduate student here at Utah State University, Dr. Fronk, Dr. Folkman, Dr. Li, Dr. Yu, Dr. Smith, and Dr. Powell.

My sincere appreciation to Dr. Thomas Fronk for helping me develop this unique thesis topic and guiding me through the analysis. It has been a wild ride and I have learned a lot.

I would also like to thank coffee, beer, and my cat, Tinkerbell.

Braydon Day Hall

# CONTENTS

# LIST OF TABLES

Table Page

LIST OF FIGURES

# LIST OF SYMBOLS

| | |
|---|---|
| $[M]$ | Mass Matrix |
| $[C]$ | Viscous Damping Matrix |
| $[K]$ | Stiffness Matrix |
| $[K]_{\Re}$ | Real Portion of Stiffness Matrix |
| $[K]_{\Im}$ | Complex Portion of Stiffness Matrix |
| $[K]_{SS}$ | Stiffness Matrix for Stainless Steel |
| $[K]_{C}$ | Stiffness Matrix for Composite Material |
| $[K]_{VE}$ | Stiffness Matrix for Viscoelastic Material |
| $[K]_{VE_{\Re}}$ | Real Portion of Stiffness Matrix for Viscoelastic Material |
| $[K]_{VE_{\Im}}$ | Complex Portion of Stiffness Matrix for Viscoelastic Material |
| $i$ | $\sqrt{-1}$ |
| $\{\ddot{x}\}$ | Acceleration Vector |
| $\{\dot{x}\}$ | Velocity Vector |
| $\{x\}$ | Displacement Vector |
| $U$ | Eigenvector |
| $U^T$ | Transpose of Eigenvector |
| $U^*$ | Complex Eigenvector |
| $U^{*T}$ | Transpose of Complex Eigenvector |
| $U_{\Re}$ | Real Portion of Eigenvector |
| $U_{\Im}$ | Complex Portion of Eigenvector |
| $\omega^*$ | Complex Frequency |
| $\omega$ | Frequency |

| | |
|---|---|
| $\omega_n$ | Natural Frequency |
| $\eta$ | Loss factor (MSE) |
| $\eta_{VE}$ | Loss Factor for Viscoelastic Material (MSE) |
| $t$ | Time |
| $V$ | Strain Energy |
| $V_{VE}$ | Strain Energy for Viscoelastic Material |
| $\mathbf{V}$ | Volume |
| $x$ | Measure in Axial Direction |
| $\theta$ | Measure in Circumferential Direction |
| $r$ | Measure in Radial Direction |
| $A$ | Area |
| $\Delta$ | Infinitesimal Change |
| $\tau$ | Shear Stress |
| $\sigma$ | Stress |
| $F$ | Force |
| $m$ | Mass |
| $a$ | Acceleration |
| $u$ | Displacement |
| $\ddot{u}$ | Acceleration |
| $\rho$ | Density |
| $\varepsilon$ | Strain |
| $\gamma$ | Shear Strain |
| $C$ | Stiffness |
| $\bar{C}$ | Transformed Stiffness |

| | |
|---|---|
| $\nu$ | Poisson's Ratio |
| $E$ | Young's Modulus |
| $G$ | Shear Modulus |
| $\psi$ | Shape Function |
| $K$ | Stiffness |
| $M$ | Mass |
| $\xi$ | Element Coordinate System in $r$ Direction |
| $\eta$ | Element Coordinate System in $r$ Direction (FEM) |
| $[J]$ | Jacobian Matrix |
| $\|J\|$ | Determinant of Jacobian Matrix |
| $[J^*]$ | Inverse of Jacobian Matrix |
| $W$ | Weight at Gauss Points |
| $t$ | Time |
| $N$ | Normal Forces (Timoshenko) |
| $M$ | Bending Moment (Timoshenko) |
| $Q$ | Shearing Forces (Timoshenko) |
| $h$ | Thickness (Timoshenko) |
| $P$ | Load (Timoshenko) |
| $\beta$ | Constant (Timoshenko) |
| $\zeta_1$ | Lower Frequency Specific Raleigh Damping Value |
| $\zeta_2$ | Upper Frequency Specific Raleigh Damping Value |
| $a_0$ | Raleigh Damping Coefficient, Lower Frequency |
| $a_1$ | Raleigh Damping Coefficient, Upper Frequency |

# CHAPTER 1

# INTRODUCTION

The thesis background and proposal are presented in this chapter. A brief description of why a composite overwrapped gun barrel was proposed to incorporate viscoelastic damping layers is given. Calculated loss factors and optimal layup design are discussed.

## 1.1 Objectives

The objective of this thesis is to investigate how constrained layer viscoelastic damping dissipates energy in composite overwrapped gun barrels. This investigation focuses on simplifying this complex problem by using the Modal Strain Energy (MSE) method to approximate energy losses and subsequently, identifying a sequence of layups that optimize energy dissipation.

## 1.2 Damping

Composite structures may be used to optimize stiffness and damping. Constrained layer damping is one way of increasing damping in a structure and is accomplished by placing a viscoelastic material in between the laminae of the composite [1]. Under loading, the shear stresses increase due to the deformation of the viscoelastic layers. The viscoelastic deformation results in an increase in strain energy and is an indicator of damping.

The Modal Strain Energy (MSE) method uses the strain energy of the viscoelastic layer and of the entire system to measure damping characteristics. The resulting loss factor is an indication of how much the system is damped. A large loss factor is directly correlated to a large energy dissipation. Therefore, maximizing interlaminar shear stresses of a viscoelastic composite structure will maximize energy dissipation in that system.

## 1.3 Approach

The parameters that affect the analysis of a viscoelsatically damped gun barrel include temperature, humidity, material properties, geometry, fiber orientation, dynamic loading, and boundary conditions. The research of this thesis is devoted to describing how different layups affect the damping of the gun barrel. Therefore, temperature and humidity are arbitrary and are assumed to remain constant through all analyses. The material properties, stainless steel barrel liner dimensions, outside diameter, composite fiber orientation, and composite laminae layer thicknesses are maintained constant. The only variable is the thickness of the viscoelastic layers and their position in the barrel layup.

The finite element analysis of the gun barrel is a two dimensional axisymmetric model written in FORTRAN. The finite element model is used to obtain the strain energies of the viscoelastic layers and of the entire system. The loss factor is calculated from this information. Given the calculated loss factor, a Raleigh damping matrix is created using estimated maximum frequencies. The Newmark-Beta method is used to generate damped transient responses and are compared to the undamped responses.

# CHAPTER 2

# LITERATURE REVIEW

Ears may be permanently damaged when exposed to noise levels above 85 decibels. Gunfire produces sound at levels between 150 and 165 decibels. With hearing tissue destruction occurring at 180 decibels, unprotected exposure to gunfire presents a serious hazard. In combat situations, many military personnel cannot simply call “time-out” in a fire fight so that they may put in hearing protection prior to discharging their weapons. The current solution to reducing firearm noise levels is to attach a metal suppression device to the muzzle that reduces noise to about 120 decibels, Musani [2] and U.S ARMY Missle Command [3]. Even with the most advanced suppressors, hearing damage may occur. To help preserve the hearing of military personnel, Innovations Plus teamed up with Firehole Composites Analysis Group to develop an M4-A1 Integral Suppressed Weapon Barrel, Figure 2.1, that effectively reduced noise in the firearm, Firehole [4]. Additional benefits of the composite wrapped barrel include: reduced weight, increased accuracy, and increased heat dissipation all while maintaining similar barrel deflections and margins of safety.

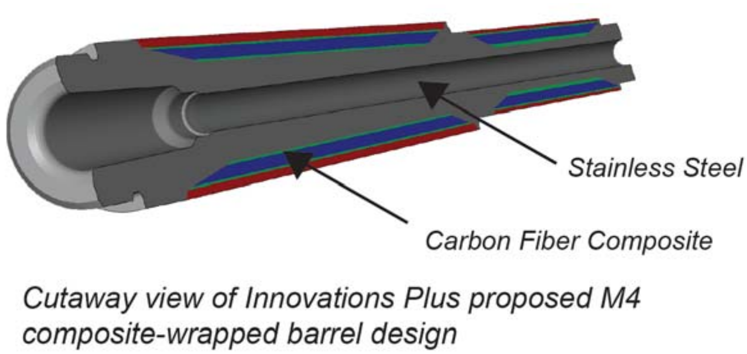



Figure 2.1 Proposed Barrel Firehole [4]

Large caliber guns have also been created for the US Navy with composite overwrap technology. Andrew Littlefield et al. have done extensive design, manufacturing, and testing on composite overwrapped 120mm gun tubes, Littlefield [5,6]. This work has resulted in gun tubes that are 205 pounds lighter than their all steel counterpart.

The differences between the coefficient of thermal expansion of metal and composite material presents difficulties in design and manufacturing. To overcome the manufacturing difficulty, a thermoplastic resin matrix was used which allowed for cure-in-place fabrication which eliminates the gap normally created due to the coefficient of thermal expansion mismatch. The pre-preg was also applied under tension resulting in a favorable pre-stress in the composite. The use of high stiffness composites helps with pointing accuracy and alleviating the dynamic strain phenomenon encountered with high-velocity projectiles.

As with all composite structures, cracks and delaminations pose a serious threat to structural integrity. When a gun fires, a stress wave travels the length of the barrel. This stress wave affects the composite-metal interface of an over wrapped barrel by increasing the shear forces. At Southern Western Mid-State University, the stress wave due to a dynamic response in a composite wrapped steel liner cylinder was modeled, Tzeng [7]. This is important because resonant stress waves can result in very high amplitude and frequency strains in the cylinder at the instant and location of the pressure front passage. A stress wave might not cause immediate failure but might accelerate an imperfection failure such as a crack or delamination. Local shell bending caused by pressure discontinuity at the pressure front as the projectile travels down tube may lead to very high axial and transverse shear stresses. The metal-composite interface response is important because the shear properties and tensile peel strength at the interface are low due to weak adhesion between composite and steel. Love's thin shell theory was used to derive a closed form expression for critical velocity for a first approximation and then the Finite Element method to simulate actual loading [8].

Numerous patents exist for carbon fiber wrapped gun barrels, each accomplishing a specific task such as reducing weight, resisting high temperatures, increasing accuracy, etc. For example, Patent number 5,657,568 is for a Composite/Metallic Gun Barrel Having a Differing, Restrictive Coefficient of Thermal Expansion [9]. This patent claims to prevent excessive expansion of metal liner in radial direction and have nearly zero coefficient of thermal expansion in axial direction.

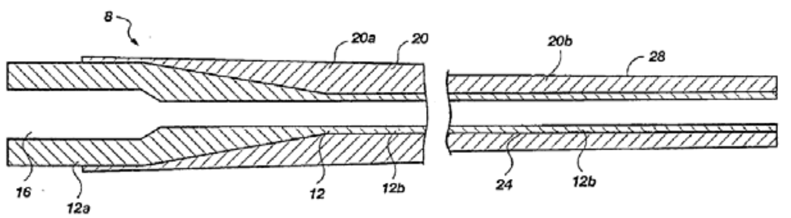



Figure 2.2 Patent 5,657,568

The moving projectile introduces a unique dynamic behavior to the mechanical response of a gun barrel. Most guns are made to be fired multiple times in succession. This translates to cyclical internal pressures and temperatures. A 3D elasticity solution was used to model such dynamic behavior in multi-layered filament-wound composite pipes [10]. The dynamic loading of repetitive fire may on cylindrical shells subjected to lateral pressure pulse loads was also studied by Khalili [11,12]. Their analysis shows that fiber layup has a significant effect on natural frequencies, transient dynamic response, and pre-stress. Also, by varying the metal volume fraction, the natural frequency may be manipulated which can be used to minimize deflection. An analysis for modeling cylindrical shells subjected to dynamic pressure was also modeled, Setoodeh [13]. Their layerwise-differential quadrature method developed for transient dynamic response is validated against the finite element method. A current review of research [14] of composite shell dynamic behavior over the past decade is presented as well as a vibrational analysis of laminated barrels [15]. This work presents equations developed for a general dynamic analysis, computing natural frequencies, non-dimensional natural frequency parameters, and an exact solution for cross-ply barrel shells. A non-dimensional model of vibration in gun barrels was presented [16]. This model shows that flexural stresses causing bending of the barrel are most influential among different types of vibration.

As the velocity of a pressure front increases to a resonance velocity behind a projectile in a barrel, stresses and strains can be amplified above what may be calculated in a static analysis. This is referred to as dynamic strain amplification [17]. Hopkins presents an analysis to help address this phenomena and its effect on creating an accurate pressure profile for use in projectile velocity calculations. Pressure gradient equations incorporating chambrage for internal ballistic codes have been developed [18]. Pressure to barrel length profiles for small calibers pressures have been calculated [19]. Pressure-Time profiles for cannons have also been created [20].

Manufacturing a composite overwrapped barrel may be done in many different ways. Each particular process presents a different method to overcoming manufacturing difficulties for a particular design constraint. Prestressing the barrel before overwrapping may be accomplished by an autofrettage process in order to overcome a coefficient of thermal expansion mismatch between composite material and metal. This is addressed by analyzing various combinations of hoop and axial winding ([0] and [0,90] layups) in order to reduce bending moments [21]. This autofrettage process has even been incorporated onto a shape memory alloy tubes reinforced with composite material [22]. Some companies have manufactured composite overwrapped barrels but provide little information on the process [23-26]. Some of these manufacturers claim their barrels dissipate heat at almost 400% faster than conventional barrels, they weigh almost 2/3 less than the same contour/length steel barrels and they balance almost every gun they're on. Additionally, they mention that a different wrap style required for a suppressed weapon. A new composite-to-metal bonding process called comelding has been bresented by The Welding Institute [27]. This utilizes a new method of modifying a metal surface to optimize bonding. Several case studies have been done that up to twelve time as much energy may be absorbed before failure than current bonding techniques. The metal altering technique known as surfi-sculpting is used to create a patterned surface for bonding. This new surface takes advantage of the shear properties of composite materials to optimize vibrational dampening by aligning the fiber direction to the metal pattern. As mentioned before, the coefficient of thermal expansion is different between metal and composites, so designing to account for this pose engineering difficulties. Making them act similar at bonding layer to prevent large stresses can be difficult.

# CHAPTER 3

# FINITE ELEMENT METHOD

## 3.1 Equations of Equilibrium

Consider a discrete representative volume of a cylinder as shown in Figure 3.1. The stress in the volume may be calculated using Newton's second law. The forces in the r, θ, and x directions are calculated by multiplying the stresses on each face of the volume by the area it is acting on and then summing them in their respective r, θ, and x directions. Higher order terms are neglected.

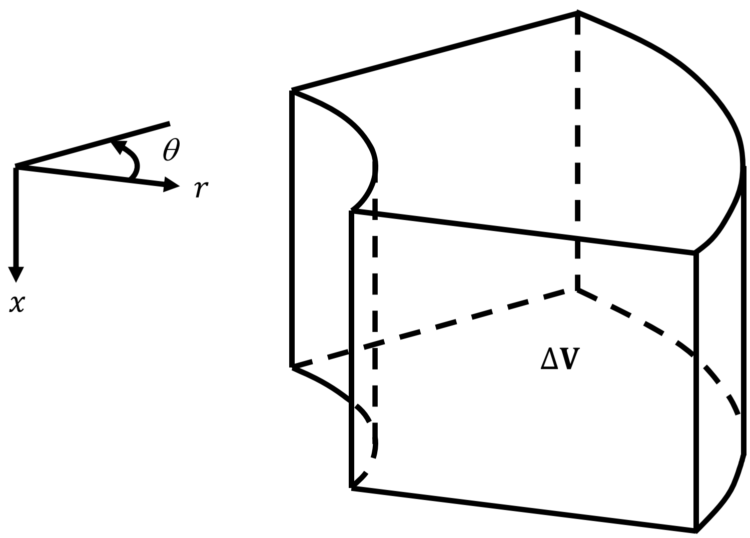



Figure 3.1 Representative Volume

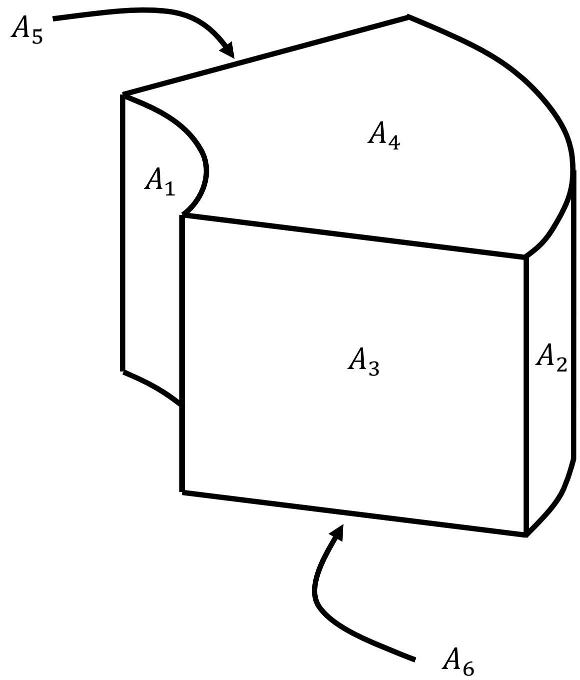



Figure 3.2 Surface Areas of Representative Volume

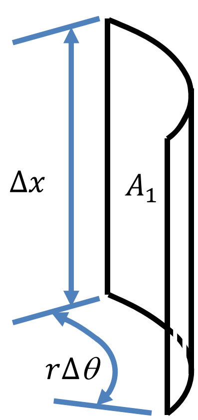



Figure 3.3 Dimensions of $A_1$

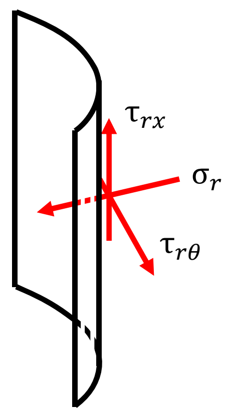



Figure 3.4 Forces on $A_1$

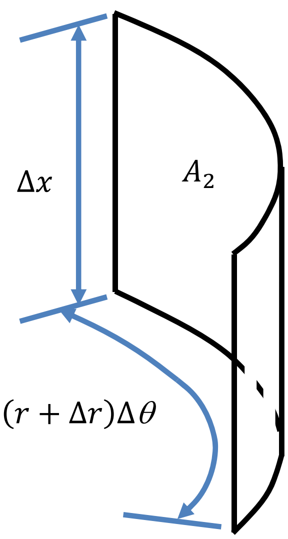



Figure 3.5 Dimensions of $A_2$

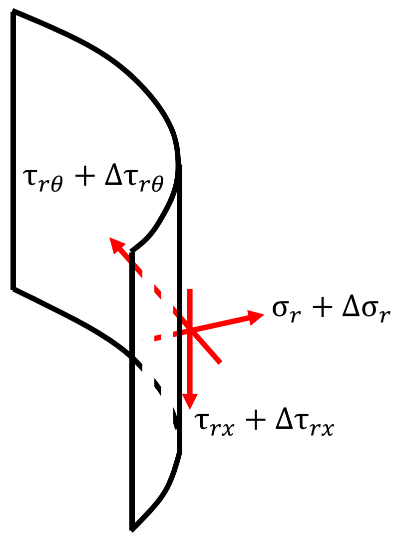



Figure 3.6 Forces on $A_2$

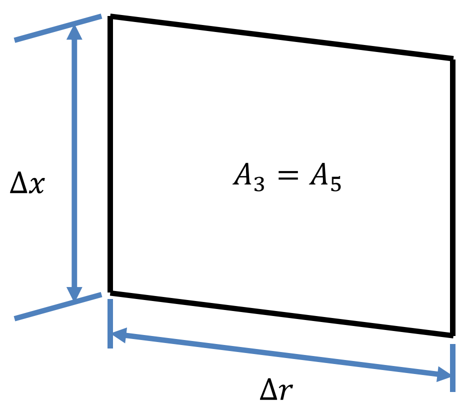



Figure 3.7 Dimensions of $A_3$ and $A_5$

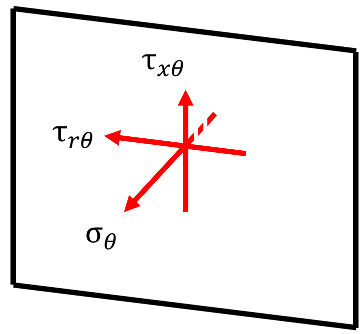



Figure 3.8 Forces on $A_3$

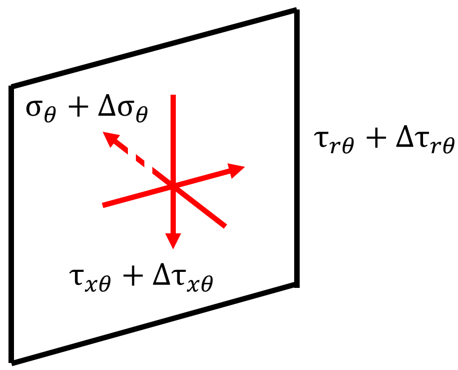



Figure 3.9 Forces on $A_5$

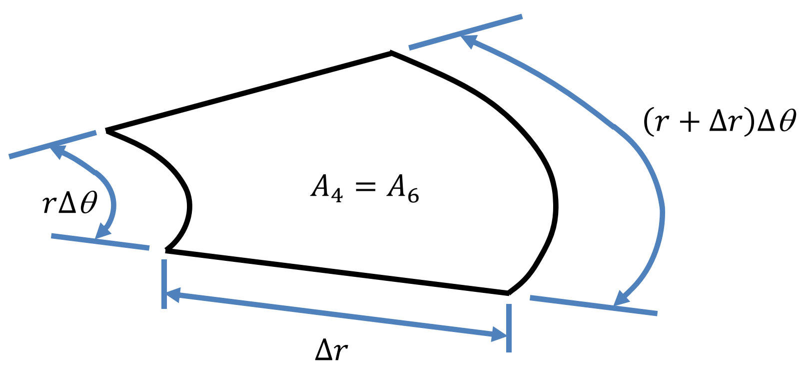



Figure 3.10 Dimensions of $A_4$ and $A_6$

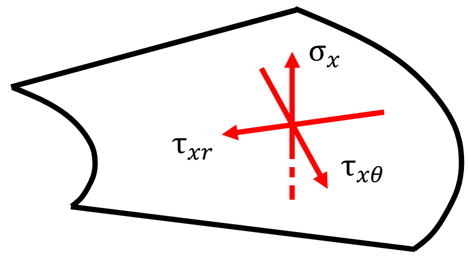



Figure 3.11 Forces on $A_4$

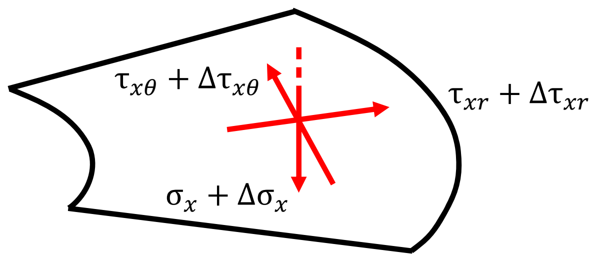



Figure 3.12 Forces on $A_6$

The surface areas and volume are,

$$A_1 = r\Delta\theta\Delta x \tag{3.1}$$

$$A_2 = r\Delta\theta\Delta x + \Delta r\Delta\theta\Delta x \tag{3.2}$$

$$A_3 = \Delta r\Delta x \tag{3.3}$$

$$A_4 = r\Delta\theta\Delta r + \frac{1}{2}\Delta\theta\Delta r^2 \tag{3.4}$$

$$A_5 = A_3 \tag{3.5}$$

$$A_6 = A_4 \tag{3.6}$$

$$\Delta\mathbf{V} = \Delta x A_4 \tag{3.7}$$

The following assumptions for small angles substituted into the force equations in the r-direction reduce the forces to,

$$\Delta r^2 = 0 \tag{3.8}$$

$$\Delta\theta^2 = 0 \tag{3.9}$$

$$\Delta x^2 = 0 \tag{3.10}$$

$$\tau_{r\theta} = \tau_{\theta r} \tag{3.11}$$

$$\tau_{x\theta} = \tau_{\theta x} \tag{3.12}$$

$$\tau_{rx} = \tau_{xr} \tag{3.13}$$

$$\Delta\tau_{r\theta} = \Delta\tau_{\theta r} \tag{3.14}$$

$$\Delta\tau_{x\theta} = \Delta\tau_{\theta x} \tag{3.15}$$

$$\Delta\tau_{rx} = \Delta\tau_{xr} \tag{3.16}$$

$$\frac{\partial\tau_{r\theta}}{\partial\theta} = \frac{\Delta\tau_{r\theta}}{\Delta\theta} \tag{3.17}$$

$$\frac{\partial\tau_{x\theta}}{\partial x} = \frac{\Delta\tau_{x\theta}}{\Delta x} \tag{3.18}$$

$$\frac{\partial\tau_{rx}}{\partial x} = \frac{\Delta\tau_{rx}}{\Delta x} \tag{3.19}$$

$$\frac{\partial\sigma_r}{\partial r} = \frac{\Delta\sigma_r}{\Delta r} \tag{3.20}$$

$$\frac{\partial\sigma_\theta}{\partial\theta} = \frac{\Delta\sigma_\theta}{\Delta\theta} \tag{3.21}$$

$$\frac{\partial\sigma_x}{\partial x} = \frac{\Delta\sigma_x}{\Delta x} \tag{3.22}$$

$$\cos\left(\frac{\Delta\theta}{2}\right) = 1 \tag{3.23}$$

$$\sin\left(\frac{\Delta\theta}{2}\right) = \frac{\Delta\theta}{2} \tag{3.24}$$

$$\cos(\Delta\theta) = 1 \tag{3.25}$$

$$\sin(\Delta\theta) = \Delta\theta \tag{3.26}$$

The forces on each face for the $r$ direction are,

$$F_{1_r} = \left(\sigma_r \cos\left(\frac{\Delta\theta}{2}\right)\right) A_1 + \left(\tau_{r\theta} \sin\left(\frac{\Delta\theta}{2}\right)\right) A_1 \tag{3.27}$$

$$F_{1_r} = \sigma_r r \Delta x \Delta\theta \tag{3.28}$$

$$F_{2_r} = \left((\sigma_r + \Delta\sigma_r) \cos\left(\frac{\Delta\theta}{2}\right)\right) A_2 + \left((\tau_{r\theta} + \Delta\tau_{r\theta}) \sin\left(\frac{\Delta\theta}{2}\right)\right) A_2 \tag{3.29}$$

$$F_{2_r} = \sigma_r r \Delta x \Delta\theta + \sigma_r \Delta r \Delta x \Delta\theta + \Delta r \Delta x \Delta\theta \frac{\partial \sigma_r}{\partial r} \tag{3.30}$$

$$F_{3_r} = (\tau_{r\theta}) A_3 \tag{3.31}$$

$$F_{3_r} = \tau_{r\theta} \Delta r \Delta x \tag{3.32}$$

$$F_{4_r} = \left((\tau_{rx} + \Delta\tau_{rx}) \cos\left(\frac{\Delta\theta}{2}\right)\right) A_4 + \left((\tau_{x\theta} + \Delta\tau_{x\theta}) \sin\left(\frac{\Delta\theta}{2}\right)\right) A_4 \tag{3.33}$$

$$F_{4_r} = \tau_{rx} r \Delta r \Delta\theta + \Delta\tau_{rx} r \Delta r \Delta\theta \tag{3.34}$$

$$F_{5_r} = (-(\sigma_\theta + \Delta\sigma_\theta) \sin(\Delta\theta)) A_5 + \left((\tau_{r\theta} + \Delta\tau_{r\theta}) \cos(\Delta\theta)\right) A_5 \tag{3.35}$$

$$F_{5_r} = \tau_{r\theta} \Delta r \Delta x - \sigma_\theta \Delta r \Delta x \Delta\theta + \Delta r \Delta x \Delta\theta \frac{\partial \tau_{r\theta}}{\partial \theta} \tag{3.36}$$

$$F_{6_r} = \left(\tau_{rx} \cos\left(\frac{\Delta\theta}{2}\right)\right) A_6 + \left(\tau_{x\theta} \sin\left(\frac{\Delta\theta}{2}\right)\right) A_6 \tag{3.37}$$

$$F_{6_r} = \tau_{rx} r \Delta r \Delta\theta \tag{3.38}$$

Summing the forces in the $r$ direction,

$$\sum F_r = m a_r \tag{3.39}$$

$$\left(F_{2_r} - F_{1_r}\right) + \left(F_{4_r} - F_{6_r}\right) + \left(F_{5_r} - F_{3_r}\right) = \rho \Delta \mathbf{V} \ddot{u}_r \tag{3.40}$$

$$\frac{\sigma_r}{r} + \frac{\partial \sigma_r}{\partial r} + \frac{\partial \tau_{rx}}{\partial x} - \frac{\sigma_\theta}{r} + \frac{1}{r} \frac{\partial \tau_{r\theta}}{\partial \theta} = \frac{\rho}{r} \ddot{u}_r \tag{3.41}$$

For axisymmetric,

$$\frac{\partial \tau_{r\theta}}{\partial \theta} = 0 \tag{3.42}$$

$$\frac{\sigma_r}{r} + \frac{\partial \sigma_r}{\partial r} + \frac{\partial \tau_{rx}}{\partial x} - \frac{\sigma_\theta}{r} = \frac{\rho}{r} \ddot{u}_r \tag{3.43}$$

For the $\theta$ direction,

$$F_{1_\theta} = \left(\tau_{r\theta} \cos\left(\frac{\Delta\theta}{2}\right)\right) A_1 + \left(\sigma_r \sin\left(\frac{\Delta\theta}{2}\right)\right) A_1 \tag{3.44}$$

$$F_{1_\theta} = \tau_{r\theta} r \Delta x \Delta\theta \tag{3.45}$$

$$F_{2_\theta} = \left((\tau_{r\theta} + \Delta\tau_{r\theta}) \cos\left(\frac{\Delta\theta}{2}\right)\right) A_2 + \left({\sigma_r}^2 \sin\left(\frac{\Delta\theta}{2}\right)\right) A_2 \tag{3.46}$$

$$F_{2_\theta} = \tau_{r\theta} r \Delta x \Delta\theta + \tau_{r\theta} \Delta r \Delta x \Delta\theta + \Delta r \Delta x \Delta\theta \Delta\tau_{r\theta} \tag{3.47}$$

$$F_{3_\theta} = (\sigma_\theta) A_3 \tag{3.48}$$

$$F_{3_\theta} = \sigma_\theta \Delta r \Delta x \tag{3.49}$$

$$F_{4_\theta} = \left((\tau_{rx} + \Delta\tau_{rx}) \sin\left(\frac{\Delta\theta}{2}\right)\right) A_4 + \left((\tau_{x\theta} + \Delta\tau_{x\theta}) \cos\left(\frac{\Delta\theta}{2}\right)\right) A_4 \tag{3.50}$$

$$F_{4_\theta} = \tau_{x\theta} r \Delta r \Delta\theta + r \Delta r \Delta\theta \Delta x \frac{\partial \tau_{x\theta}}{\partial x} \tag{3.51}$$

$$F_{5_\theta} = \big((\sigma_\theta + \Delta\sigma_\theta) \cos(\Delta\theta)\big) A_5 + \big((\tau_{r\theta} + \Delta\tau_{r\theta}) \sin(\Delta\theta)\big) A_5 \tag{3.52}$$

$$F_{5_\theta} = \sigma_\theta \Delta r \Delta x + \tau_{r\theta} \Delta r \Delta x \Delta\theta + \Delta r \Delta x \Delta\theta \frac{\partial \sigma_\theta}{\partial \theta} \tag{3.53}$$

$$F_{6_\theta} = \left(\tau_{rx} \sin\left(\frac{\Delta\theta}{2}\right)\right) A_6 + \left(\tau_{x\theta} \cos\left(\frac{\Delta\theta}{2}\right)\right) A_6 \tag{3.54}$$

$$F_{6_\theta} = \tau_{x\theta} r \Delta r \Delta\theta \tag{3.55}$$

Summing the forces in the $\theta$ direction,

$$\sum F_\theta = m a_\theta \tag{3.56}$$

$$\big(F_{2_\theta} - F_{1_\theta}\big) + \big(F_{4_\theta} - F_{6_\theta}\big) + \big(F_{5_\theta} - F_{3_\theta}\big) = \rho \Delta \mathbf{V} \ddot{u}_\theta \tag{3.57}$$

$$2\frac{\tau_{r\theta}}{r}+\frac{\partial\tau_{r\theta}}{\partial r}+\frac{\partial\tau_{x\theta}}{\partial x}+\frac{1}{r}\frac{\partial\sigma_{\theta}}{\partial\theta}=\frac{\rho}{r}\ddot{u}_{\theta} \tag{3.58}$$

For axisymmetric,

$$\frac{\partial\sigma_{\theta}}{\partial\theta}=0 \tag{3.59}$$

$$2\frac{\tau_{r\theta}}{r}+\frac{\partial\tau_{r\theta}}{\partial r}+\frac{\partial\tau_{x\theta}}{\partial x}=\frac{\rho}{r}\ddot{u}_{\theta} \tag{3.60}$$

For the $x$ direction,

$$F_{1_x}=(\tau_{rx})A_1 \tag{3.61}$$

$$F_{1_x}=\tau_{rx}r\Delta x\Delta\theta \tag{3.62}$$

$$F_{2_x}=(\tau_{rx}+\Delta\tau_{rx})A_2 \tag{3.63}$$

$$F_{2_x}=\tau_{rx}r\Delta x\Delta\theta+\tau_{rx}\Delta r\Delta x\Delta\theta+\Delta r\Delta x\Delta\theta\frac{\partial\tau_{rx}}{\partial r} \tag{3.64}$$

$$F_{3_x}=(\tau_{x\theta})A_3 \tag{3.65}$$

$$F_{3_x}=\tau_{x\theta}\Delta r\Delta x \tag{3.66}$$

$$F_{4_x}=(\sigma_x)A_4 \tag{3.67}$$

$$F_{4_x}=\sigma_x r\Delta r\Delta\theta \tag{3.68}$$

$$F_{5_x}=(\tau_{x\theta}+\Delta\tau_{x\theta})A_5 \tag{3.69}$$

$$F_{5_x}=\tau_{x\theta}\Delta r\Delta x+\Delta r\Delta x\Delta\theta\frac{\partial\tau_{x\theta}}{\partial\theta} \tag{3.70}$$

$$F_{6_x}=(\sigma_x+\Delta\sigma_x)A_6 \tag{3.71}$$

$$F_{6_x}=\sigma_x r\Delta r\Delta\theta+r\Delta r\Delta x\Delta\theta\frac{\partial\sigma_x}{\partial x} \tag{3.72}$$

Summing the forces in the $x$ direction,

$$\sum F_x=ma_x \tag{3.73}$$

$$\left(F_{2_x} - F_{1_x}\right) + \left(F_{6_x} - F_{4_x}\right) + \left(F_{5_x} - F_{3_x}\right) = \rho \Delta \mathbf{V} \ddot{u}_x \tag{3.74}$$

$$\frac{\tau_{rx}}{r} + \frac{\partial \tau_{rx}}{\partial r} + \frac{\partial \sigma_x}{\partial x} + \frac{1}{r}\frac{\partial \tau_{x\theta}}{\partial \theta} = \frac{\rho}{r}\ddot{u}_x \tag{3.75}$$

For axisymmetric,

$$\frac{\partial \tau_{x\theta}}{\partial \theta} = 0 \tag{3.76}$$

$$\frac{\tau_{rx}}{r} + \frac{\partial \tau_{rx}}{\partial r} + \frac{\partial \sigma_x}{\partial x} = \frac{\rho}{r}\ddot{u}_x \tag{3.77}$$

In summary,

$$\frac{\sigma_r}{r} + \frac{\partial \sigma_r}{\partial r} + \frac{\partial \tau_{rx}}{\partial x} - \frac{\sigma_\theta}{r} = \frac{\rho}{r}\ddot{u}_r \tag{3.43}$$

$$2\frac{\tau_{r\theta}}{r} + \frac{\partial \tau_{r\theta}}{\partial r} + \frac{\partial \tau_{x\theta}}{\partial x} = \frac{\rho}{r}\ddot{u}_\theta \tag{3.60}$$

$$\frac{\tau_{rx}}{r} + \frac{\partial \tau_{rx}}{\partial r} + \frac{\partial \sigma_x}{\partial x} = \frac{\rho}{r}\ddot{u}_x \tag{3.77}$$

## 3.2 Kinematic Equations

The kinematic equations are defined as,

$$\varepsilon_x = \frac{du_x}{dx} \tag{3.78}$$

$$\varepsilon_\theta = \frac{u_r}{r} + \frac{1}{r}\frac{du_\theta}{d\theta} \tag{3.79}$$

$$\varepsilon_r = \frac{du_r}{dr} \tag{3.80}$$

$$\gamma_{\theta r} = \frac{1}{r}\left(\frac{du_r}{d\theta} + r\frac{du_\theta}{dr} - u_\theta\right) \tag{3.81}$$

$$\gamma_{rx} = \frac{du_x}{dr} + \frac{du_r}{dx} \tag{3.82}$$

$$\gamma_{x\theta} = \frac{du_\theta}{dx} + \frac{1}{r}\frac{du_x}{d\theta} \tag{3.83}$$

For axisymmetric,

$$\frac{du_\theta}{d\theta} = 0 \tag{3.84}$$

$$\frac{du_x}{d\theta} = 0 \tag{3.85}$$

$$\frac{du_r}{d\theta} = 0 \tag{3.86}$$

Therefore,

$$\varepsilon_\theta = \frac{u_r}{r} \tag{3.87}$$

$$\gamma_{\theta r} = \frac{du_\theta}{dr} - \frac{u_\theta}{r} \tag{3.88}$$

$$\gamma_{x\theta} = \frac{du_\theta}{dx} \tag{3.89}$$

## 3.3 Constitutive Equations

Assuming anisotropic material, the constitutive equations [28] are defined as,

$$\begin{Bmatrix} \sigma_x \\ \sigma_\theta \\ \sigma_r \\ \tau_{\theta r} \\ \tau_{rx} \\ \tau_{x\theta} \end{Bmatrix} = \begin{bmatrix} \bar{C}_{11} & \bar{C}_{12} & \bar{C}_{13} & 0 & 0 & \bar{C}_{16} \\ \bar{C}_{21} & \bar{C}_{22} & \bar{C}_{23} & 0 & 0 & \bar{C}_{26} \\ \bar{C}_{31} & \bar{C}_{32} & \bar{C}_{33} & 0 & 0 & \bar{C}_{36} \\ 0 & 0 & 0 & \bar{C}_{44} & \bar{C}_{45} & 0 \\ 0 & 0 & 0 & \bar{C}_{54} & \bar{C}_{55} & 0 \\ \bar{C}_{61} & \bar{C}_{62} & \bar{C}_{63} & 0 & 0 & \bar{C}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_\theta \\ \varepsilon_r \\ \gamma_{\theta r} \\ \gamma_{rx} \\ \gamma_{x\theta} \end{Bmatrix} \tag{3.90}$$

Where the transformed stiffness values are defined in terms of the principal stiffness terms as,

$$\bar{C}_{11} = C_{11}(\cos\theta)^4 + 2(C_{12} + 2C_{66})(\cos\theta)^2(\sin\theta)^2 + C_{22}(\sin\theta)^4 \tag{3.91}$$

$$\bar{C}_{12} = (C_{11} + C_{12} - 4C_{66})(\cos\theta)^2(\sin\theta)^2 + C_{12}((\cos\theta)^4 + (\sin\theta)^4) \tag{3.92}$$

$$\bar{C}_{13} = C_{13}(\cos\theta)^2 + C_{23}(\sin\theta)^2 \tag{3.93}$$

$$\begin{aligned}\bar{C}_{16} = \big(&C_{11}(\cos\theta)^2 - C_{22}(\sin\theta)^2 \\ &- (C_{12} + 2C_{66})((\cos\theta)^2 - (\sin\theta)^2)\big)\cos\theta\sin\theta\end{aligned} \tag{3.94}$$

$$\bar{C}_{22} = C_{11}(\sin\theta)^4 + 2(C_{12} + 2C_{66})(\cos\theta)^2(\sin\theta)^2 + C_{22}(\cos\theta)^4 \tag{3.95}$$

$$\bar{C}_{23} = C_{13}(\sin\theta)^2 + C_{23}(\cos\theta)^2 \tag{3.96}$$

$$\bar{C}_{33} = C_{33} \tag{3.97}$$

$$\begin{aligned}\bar{C}_{26} = \big(&C_{11}(\sin\theta)^2 - C_{22}(\cos\theta)^2 \\ &+ (C_{12} + 2C_{66})((\cos\theta)^2 - (\sin\theta)^2)\big)\cos\theta\sin\theta\end{aligned} \tag{3.98}$$

$$\bar{C}_{36} = (C_{13} - C_{23})\sin\theta\cos\theta \tag{3.99}$$

$$\bar{C}_{44} = C_{55}(\sin\theta)^2 + C_{44}(\cos\theta)^2 \tag{3.100}$$

$$\bar{C}_{45} = (C_{55} - C_{44})\sin\theta\cos\theta \tag{3.101}$$

$$\bar{C}_{55} = C_{44}(\sin\theta)^2 + C_{55}(\cos\theta)^2 \tag{3.102}$$

$$\bar{C}_{66} = (C_{11} + C_{12} - 2C_{12})(\cos\theta)^2(\sin\theta)^2 + C_{66}((\cos\theta)^2 - (\sin\theta)^2)^2 \tag{3.103}$$

The principal stiffness values are defined as,

$$C_{11} = \frac{(1 - \nu_{23}\nu_{32})E_1}{1 - \nu} \tag{3.104}$$

$$C_{12} = \frac{(\nu_{21} + \nu_{31}\nu_{23})E_1}{1 - \nu} = \frac{(\nu_{12} + \nu_{13}\nu_{32})E_2}{1 - \nu} \tag{3.105}$$

$$C_{13} = \frac{(\nu_{31} + \nu_{21}\nu_{32})E_1}{1 - \nu} = \frac{(\nu_{13} + \nu_{12}\nu_{23})E_3}{1 - \nu} \tag{3.106}$$

$$C_{22} = \frac{(1 - \nu_{13}\nu_{31})E_2}{1 - \nu} \tag{3.107}$$

$$C_{23} = \frac{(\nu_{32} + \nu_{12}\nu_{31})E_2}{1 - \nu} = \frac{(\nu_{23} + \nu_{21}\nu_{13})E_3}{1 - \nu} \tag{3.108}$$

$$C_{33} = \frac{(1 - \nu_{12}\nu_{21})E_3}{1 - \nu} \tag{3.109}$$

$$C_{44} = G_{23} \tag{3.110}$$

$$C_{55} = G_{13} \tag{3.111}$$

$$C_{66} = G_{12} \tag{3.112}$$

$$\nu = \nu_{12}\nu_{21} + \nu_{23}\nu_{32} + \nu_{31}\nu_{13} + 2\nu_{21}\nu_{32}\nu_{13} \tag{3.113}$$

The principal constitutive equations,

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} \tag{3.114}$$

For axisymmetric,

$$\bar{C}_{ij} = \bar{C}_{ji} \tag{3.115}$$

$$C_{ij} = C_{ji} \tag{3.116}$$

Expanding the matrix into six equations,

$$\sigma_x = \bar{C}_{11}\varepsilon_x + \bar{C}_{12}\varepsilon_\theta + \bar{C}_{13}\varepsilon_r + \bar{C}_{16}\gamma_{x\theta} \tag{3.117}$$

$$\sigma_\theta = \bar{C}_{12}\varepsilon_x + \bar{C}_{22}\varepsilon_\theta + \bar{C}_{23}\varepsilon_r + \bar{C}_{26}\gamma_{x\theta} \tag{3.118}$$

$$\sigma_r = \bar{C}_{13}\varepsilon_x + \bar{C}_{23}\varepsilon_\theta + \bar{C}_{33}\varepsilon_r + \bar{C}_{36}\gamma_{x\theta} \tag{3.119}$$

$$\tau_{\theta r} = \bar{C}_{44}\gamma_{\theta r} + \bar{C}_{45}\gamma_{rx} \tag{3.120}$$

$$\tau_{rx} = \bar{C}_{45}\gamma_{\theta r} + \bar{C}_{55}\gamma_{rx} \tag{3.121}$$

$$\tau_{x\theta} = \bar{C}_{16}\varepsilon_x + \bar{C}_{26}\varepsilon_\theta + \bar{C}_{36}\varepsilon_r + \bar{C}_{66}\gamma_{x\theta} \tag{3.122}$$

Substituting Equations (3.78), (3.80), (3.82), (3.87), (3.88), and (3.89) into (3.117), through (3.122),

$$\sigma_x = \bar{C}_{11}\frac{du_x}{dx} + \bar{C}_{12}\frac{u_r}{r} + \bar{C}_{13}\frac{du_r}{dr} + \bar{C}_{16}\frac{du_\theta}{dx} \tag{3.123}$$

$$\frac{\partial\sigma_x}{\partial x} = \frac{\partial}{\partial x}\left(\bar{C}_{11}\frac{du_x}{dx} + \bar{C}_{12}\frac{u_r}{r} + \bar{C}_{13}\frac{du_r}{dr} + \bar{C}_{16}\frac{du_\theta}{dx}\right) \tag{3.124}$$

$$\sigma_\theta = \bar{C}_{12}\frac{du_x}{dx} + \bar{C}_{22}\frac{u_r}{r} + \bar{C}_{23}\frac{du_r}{dr} + \bar{C}_{26}\frac{du_\theta}{dx} \tag{3.125}$$

$$\frac{\partial\sigma_\theta}{\partial \theta} = \frac{\partial}{\partial \theta}\left(\bar{C}_{12}\frac{du_x}{dx} + \bar{C}_{22}\frac{u_r}{r} + \bar{C}_{23}\frac{du_r}{dr} + \bar{C}_{26}\frac{du_\theta}{dx}\right) \tag{3.126}$$

$$\sigma_r = \bar{C}_{13}\frac{du_x}{dx} + \bar{C}_{23}\frac{u_r}{r} + \bar{C}_{33}\frac{du_r}{dr} + \bar{C}_{36}\frac{du_\theta}{dx} \tag{3.127}$$

$$\frac{\partial\sigma_r}{\partial r} = \frac{\partial}{\partial r}\left(\bar{C}_{13}\frac{du_x}{dx} + \bar{C}_{23}\frac{u_r}{r} + \bar{C}_{33}\frac{du_r}{dr} + \bar{C}_{36}\frac{du_\theta}{dx}\right) \tag{3.128}$$

$$\tau_{\theta r} = \bar{C}_{44}\gamma_{\theta r} + \bar{C}_{45}\gamma_{rx} \tag{3.129}$$

$$\frac{\partial\tau_{\theta r}}{\partial r} = \frac{\partial}{\partial r}(\bar{C}_{44}\gamma_{\theta r} + \bar{C}_{45}\gamma_{rx}) \tag{3.130}$$

$$\frac{\partial\tau_{\theta r}}{\partial \theta} = \frac{\partial}{\partial \theta}(\bar{C}_{44}\gamma_{\theta r} + \bar{C}_{45}\gamma_{rx}) \tag{3.131}$$

$$\tau_{rx} = \bar{C}_{45}\gamma_{\theta r} + \bar{C}_{55}\gamma_{rx} \tag{3.132}$$

$$\frac{\partial\tau_{rx}}{\partial r} = \frac{\partial}{\partial r}(\bar{C}_{45}\gamma_{\theta r} + \bar{C}_{55}\gamma_{rx}) \tag{3.133}$$

$$\frac{\partial\tau_{rx}}{\partial x} = \frac{\partial}{\partial x}(\bar{C}_{45}\gamma_{\theta r} + \bar{C}_{55}\gamma_{rx}) \tag{3.134}$$

$$\tau_{x\theta} = \bar{C}_{16}\frac{du_x}{dx} + \bar{C}_{26}\frac{u_r}{r} + \bar{C}_{36}\frac{du_r}{dr} + \bar{C}_{66}\frac{du_\theta}{dx} \tag{3.135}$$

$$\frac{\partial\tau_{x\theta}}{\partial \theta} = \frac{\partial}{\partial \theta}\left(\bar{C}_{16}\frac{du_x}{dx} + \bar{C}_{26}\frac{u_r}{r} + \bar{C}_{36}\frac{du_r}{dr} + \bar{C}_{66}\frac{du_\theta}{dx}\right) \tag{3.136}$$

$$\frac{\partial\tau_{x\theta}}{\partial x} = \frac{\partial}{\partial x}\left(\bar{C}_{16}\frac{du_x}{dx} + \bar{C}_{26}\frac{u_r}{r} + \bar{C}_{36}\frac{du_r}{dr} + \bar{C}_{66}\frac{du_\theta}{dx}\right) \tag{3.137}$$

## 3.4 Stiffness and Mass Matrix Derivation

Substitute equations (3.123) through (3.137) into (3.43), (3.60), and (3.77). Using variational calculus, the stiffness and mass matrix equations may be derived. The following demonstrates this derivation for the $r$ direction. Beginning with equation (3.43), applying the variational form, and $i$ representing the nodes per element,

$$\int_V \psi_i \left(\frac{\sigma_r}{r} + \frac{\partial \sigma_r}{\partial r} + \frac{\partial \tau_{rx}}{\partial x} - \frac{\sigma_\theta}{r}\right) dV = \int_V \psi_i \frac{\rho}{r} \ddot{u}_r \, d\mathbf{V} \tag{3.138}$$

$$\iiint \psi_i \left(\frac{\sigma_r}{r} + \frac{\partial \sigma_r}{\partial r} + \frac{\partial \tau_{rx}}{\partial x} - \frac{\sigma_\theta}{r}\right) r dr dx d\theta = \iiint \psi_i \rho \ddot{u}_r dr dx d\theta \tag{3.139}$$

$$2\pi \iint \psi_i \left(\frac{\sigma_r}{r} + \frac{\partial \sigma_r}{\partial r} + \frac{\partial \tau_{rx}}{\partial x} - \frac{\sigma_\theta}{r}\right) r dr dx = 2\pi \iint \psi_i \rho \ddot{u}_r dr dx \tag{3.140}$$

$$\begin{gathered} 2\pi \iint \psi_i (\sigma_r - \sigma_\theta) dr dx + 2\pi \iint \psi_i \left(\frac{\partial \sigma_r}{\partial r}\right) r dr dx + 2\pi \iint \psi_i \left(\frac{\partial \tau_{rx}}{\partial x}\right) r dr dx \\ = 2\pi \iint \psi_i \rho \ddot{u}_r dr dx \end{gathered} \tag{3.141}$$

Noting that,

$$\psi_i \left(\frac{\partial \sigma_r}{\partial r}\right) r = \psi_i \left(\frac{\partial}{\partial r}(\sigma_r r) - \sigma_r\right) \tag{3.142}$$

$$\frac{\partial \psi_i}{\partial r} \sigma_r r = \frac{\partial}{\partial r}(\psi_i r) \sigma_r - \psi_i \sigma_r \tag{3.143}$$

Substituting and integrating by parts,

$$\begin{gathered} -2\pi \iint \left(\psi_i \sigma_\theta + \sigma_r r \frac{\partial \psi_i}{\partial r} + \tau_{rx} r \frac{\partial \psi_i}{\partial x}\right) dr dx + 2\pi \int \psi_i \sigma_r r dx + 2\pi \int \psi_i \tau_{rx} r dr \\ = 2\pi \iint \psi_i \rho \ddot{u}_r dr dx \end{gathered} \tag{3.144}$$

Substituting equations (3.125), (3.127), and (3.132) and regrouping in terms of $u_x$, $u_r$, $u_\theta$, $du_x$, $du_r$, and $du_\theta$,

$$-2\pi\iint\left[u_r\left(\frac{\bar{C}_{22}}{r}\psi_i+\bar{C}_{23}\frac{\partial\psi_i}{\partial r}\right)-u_\theta\left(\bar{C}_{45}\frac{\partial\psi_i}{\partial x}\right)+\frac{du_r}{dr}\left(\bar{C}_{23}\psi_i+r\bar{C}_{33}\frac{\partial\psi_i}{\partial r}\right)\right.$$
$$+\frac{du_r}{dx}\left(r\bar{C}_{55}\frac{\partial\psi_i}{\partial x}\right)+\frac{du_\theta}{dr}\left(r\bar{C}_{45}\frac{\partial\psi_i}{\partial x}\right)$$
$$+\frac{du_\theta}{dx}\left(\bar{C}_{26}\psi_i+r\bar{C}_{36}\frac{\partial\psi_i}{\partial r}\right)+\frac{du_x}{dr}\left(r\bar{C}_{55}\frac{\partial\psi_i}{\partial x}\right) \tag{3.145}$$
$$\left.+\frac{du_x}{dx}\left(\bar{C}_{12}\psi_i+r\bar{C}_{13}\frac{\partial\psi_i}{\partial r}\right)\right]drdx+2\pi\int\psi_i\sigma_r rdx$$
$$+2\pi\int\psi_i\tau_{rx}rdr=2\pi\iint\psi_i\rho\ddot{u}_r drdx$$

Substitute in isoparametric elements,

$$u_r=\sum_{j=1}^{NPE}u_{r\,j}\psi_j \tag{3.146}$$

$$u_\theta=\sum_{j=1}^{NPE}u_{\theta\,j}\psi_j \tag{3.147}$$

$$u_x=\sum_{j=1}^{NPE}u_{x\,j}\psi_j \tag{3.148}$$

$$\frac{du_r}{dr}=\sum_{j=1}^{NPE}u_{r\,j}\frac{d\psi_j}{dr} \tag{3.149}$$

$$\frac{du_\theta}{dr}=\sum_{j=1}^{NPE}u_{\theta\,j}\frac{d\psi_j}{dr} \tag{3.150}$$

$$\frac{du_x}{dr}=\sum_{j=1}^{NPE}u_{x\,j}\frac{d\psi_j}{dr} \tag{3.151}$$

$$\frac{du_r}{dx}=\sum_{j=1}^{NPE}u_{r\,j}\frac{d\psi_j}{dx} \tag{3.152}$$

$$\frac{du_\theta}{dx} = \sum_{j=1}^{NPE} u_{\theta\,j} \frac{d\psi_j}{dx} \tag{3.153}$$

$$\frac{du_x}{dx} = \sum_{j=1}^{NPE} u_{x\,j} \frac{d\psi_j}{dx} \tag{3.154}$$

$$\ddot{u}_r = \sum_{j=1}^{NPE} \ddot{u}_{r\,j} \psi_j \tag{3.155}$$

$$\ddot{u}_\theta = \sum_{j=1}^{NPE} \ddot{u}_{\theta\,j} \psi_j \tag{3.156}$$

$$\ddot{u}_x = \sum_{j=1}^{NPE} \ddot{u}_{x\,j} \psi_j \tag{3.157}$$

The subscript, $j$, represents the node local ID in each element. The summation is taken over the total number of nodes per each element, $NPE$. Regrouping in terms of $u_{r\,j}$, $u_{\theta\,j}$, and $u_{x\,j}$,

$$\begin{aligned}
-2\pi \sum_{j=1}^{NPE} \iint \Bigg[ & u_{r\,j} \left( r\bar{C}_{45} \frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r} + \bar{C}_{26}\psi_i \frac{\partial\psi_j}{\partial x} + r\bar{C}_{36} \frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x} - \bar{C}_{45} \frac{\partial\psi_i}{\partial x}\psi_j \right) \\
& + u_{\theta\,j} \left( r\bar{C}_{45} \frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r} + \bar{C}_{26}\psi_i \frac{\partial\psi_j}{\partial x} + r\bar{C}_{36} \frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x} \right. \\
& \left. - \bar{C}_{45} \frac{\partial\psi_i}{\partial x}\psi_j \right) \\
& + u_{x\,j} \left( \bar{C}_{12}\psi_i \frac{\partial\psi_j}{\partial x} + r\bar{C}_{13} \frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x} + r\bar{C}_{55} \frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r} \right) \Bigg] drdx \\
& + 2\pi \int \psi_i \sigma_r r dx + 2\pi \int \psi_i \tau_{rx} r dr = 2\pi \sum_{j=1}^{NPE} \iint u_{r\,j} \rho \psi_i \psi_j drdx
\end{aligned} \tag{3.158}$$

Note that the equation for each $i$ th node is,

$$\sum_{j=1}^{NPE}\left[u_{x\,j}\left(K_{ij}^{21}\right)+u_{r\,j}\left(K_{ij}^{22}\right)+u_{\theta\,j}\left(K_{ij}^{23}\right)\right]+2\pi\int\psi_i\sigma_r r dx+2\pi\int\psi_i\tau_{rx} r dr$$
$$=\sum_{j=1}^{NPE}u_{r\,j}M_{ij}^{22} \tag{3.159}$$

Extracting the stiffness terms,

$$K_{ij}^{21}=-2\pi\iint\left(\bar{C}_{12}\psi_i\frac{\partial\psi_j}{\partial x}+r\bar{C}_{13}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x}+r\bar{C}_{55}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r}\right)drdx \tag{3.160}$$

$$K_{ij}^{22}=-2\pi\iint\left(\frac{\bar{C}_{22}}{r}\psi_i\psi_j+\bar{C}_{23}\frac{\partial\psi_i}{\partial r}\psi_j+\bar{C}_{23}\psi_i\frac{\partial\psi_j}{\partial r}+r\bar{C}_{33}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial r}\right.$$
$$\left.+r\bar{C}_{55}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial x}\right)drdx \tag{3.161}$$

$$K_{ij}^{23}=-2\pi\iint\left(r\bar{C}_{45}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r}+\bar{C}_{26}\psi_i\frac{\partial\psi_j}{\partial x}+r\bar{C}_{36}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x}\right.$$
$$\left.-\bar{C}_{45}\frac{\partial\psi_i}{\partial x}\psi_j\right)drdx \tag{3.162}$$

$$M_{ij}^{22}=2\pi\iint\left(\rho\psi_i\psi_j\right)drdx \tag{3.163}$$

Similarly for the $x$ and $\theta$ directions,

$$K_{ij}^{11}=-2\pi\iint\left(r\bar{C}_{55}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial r}+r\bar{C}_{11}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial x}\right)drdx \tag{3.164}$$

$$K_{ij}^{12}=-2\pi\iint\left(\bar{C}_{12}\frac{\partial\psi_i}{\partial x}\psi_j+r\bar{C}_{13}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r}+r\bar{C}_{55}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x}\right)drdx \tag{3.165}$$

$$K_{ij}^{13}=-2\pi\iint\left(r\bar{C}_{16}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial x}+r\bar{C}_{45}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial r}-\bar{C}_{45}\frac{\partial\psi_i}{\partial r}\psi_j\right)drdx \tag{3.166}$$

$$K_{ij}^{31}=-2\pi\iint\left(r\bar{C}_{16}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial x}+r\bar{C}_{45}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial r}-\bar{C}_{45}\psi_i\frac{\partial\psi_j}{\partial r}\right)drdx \tag{3.167}$$

$$K_{ij}^{32} = -2\pi \iint \left(\bar{C}_{26}\frac{\partial\psi_i}{\partial x}\psi_j + r\bar{C}_{36}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial r} + r\bar{C}_{45}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial x} - \bar{C}_{45}\psi_i\frac{\partial\psi_j}{\partial x}\right) drdx \tag{3.168}$$

$$K_{ij}^{33} = -2\pi \iint \left(r\bar{C}_{66}\frac{\partial\psi_i}{\partial x}\frac{\partial\psi_j}{\partial x} + r\bar{C}_{44}\frac{\partial\psi_i}{\partial r}\frac{\partial\psi_j}{\partial r} - \bar{C}_{44}\frac{\partial\psi_i}{\partial r}\psi_j - \bar{C}_{44}\psi_i\frac{\partial\psi_j}{\partial r} + \frac{\bar{C}_{44}}{r}\psi_i\psi_j\right) drdx \tag{3.169}$$

$$M_{ij}^{11} = 2\pi \iint \left(\rho\psi_i\psi_j\right) drdx \tag{3.170}$$

$$M_{ij}^{33} = 2\pi \iint \left(\rho\psi_i\psi_j\right) drdx \tag{3.171}$$

The resulting matrix equation is,

$$\begin{Bmatrix} F_{x\,j} \\ F_{r\,j} \\ F_{\theta\,j} \end{Bmatrix} = \begin{bmatrix} K_{ij}^{11} & K_{ij}^{12} & K_{ij}^{13} \\ K_{ij}^{21} & K_{ij}^{22} & K_{ij}^{23} \\ K_{ij}^{31} & K_{ij}^{32} & K_{ij}^{33} \end{bmatrix} \begin{Bmatrix} u_{x\,j} \\ u_{r\,j} \\ u_{\theta\,j} \end{Bmatrix} + \begin{bmatrix} M_{ij}^{11} & 0 & 0 \\ 0 & M_{ij}^{22} & 0 \\ 0 & 0 & M_{ij}^{33} \end{bmatrix} \begin{Bmatrix} \ddot{u}_{x\,j} \\ \ddot{u}_{r\,j} \\ \ddot{u}_{\theta\,j} \end{Bmatrix} \tag{3.172}$$

For a Q4 element, the stiffness and mass matrices for each element expand into to a 12x12 matrix. The terms are grouped by degree of freedom,

$$\begin{bmatrix} \begin{bmatrix} K_{11}^{11} & K_{12}^{11} & K_{13}^{11} & K_{14}^{11} \\ K_{21}^{11} & K_{22}^{11} & K_{23}^{11} & K_{24}^{11} \\ K_{31}^{11} & K_{32}^{11} & K_{33}^{11} & K_{34}^{11} \\ K_{41}^{11} & K_{42}^{11} & K_{43}^{11} & K_{44}^{11} \end{bmatrix} & \begin{bmatrix} K_{11}^{12} & K_{12}^{12} & K_{13}^{12} & K_{14}^{12} \\ K_{21}^{12} & K_{22}^{12} & K_{23}^{12} & K_{24}^{12} \\ K_{31}^{12} & K_{32}^{12} & K_{33}^{12} & K_{34}^{12} \\ K_{41}^{12} & K_{42}^{12} & K_{43}^{12} & K_{44}^{12} \end{bmatrix} & \begin{bmatrix} K_{11}^{13} & K_{12}^{13} & K_{13}^{13} & K_{14}^{13} \\ K_{21}^{13} & K_{22}^{13} & K_{23}^{13} & K_{24}^{13} \\ K_{31}^{13} & K_{32}^{13} & K_{33}^{13} & K_{34}^{13} \\ K_{41}^{13} & K_{42}^{13} & K_{43}^{13} & K_{44}^{13} \end{bmatrix} \\ \begin{bmatrix} K_{11}^{21} & K_{12}^{21} & K_{13}^{21} & K_{14}^{21} \\ K_{21}^{21} & K_{22}^{21} & K_{23}^{21} & K_{24}^{21} \\ K_{31}^{21} & K_{32}^{21} & K_{33}^{21} & K_{34}^{21} \\ K_{41}^{21} & K_{42}^{21} & K_{43}^{21} & K_{44}^{21} \end{bmatrix} & \begin{bmatrix} K_{11}^{22} & K_{12}^{22} & K_{13}^{22} & K_{14}^{22} \\ K_{21}^{22} & K_{22}^{22} & K_{23}^{22} & K_{24}^{22} \\ K_{31}^{22} & K_{32}^{22} & K_{33}^{22} & K_{34}^{22} \\ K_{41}^{22} & K_{42}^{22} & K_{43}^{22} & K_{44}^{22} \end{bmatrix} & \begin{bmatrix} K_{11}^{23} & K_{12}^{23} & K_{13}^{23} & K_{14}^{23} \\ K_{21}^{23} & K_{22}^{23} & K_{23}^{23} & K_{24}^{23} \\ K_{31}^{23} & K_{32}^{23} & K_{33}^{23} & K_{34}^{23} \\ K_{41}^{23} & K_{42}^{23} & K_{43}^{23} & K_{44}^{23} \end{bmatrix} \\ \begin{bmatrix} K_{11}^{31} & K_{12}^{31} & K_{13}^{31} & K_{14}^{31} \\ K_{21}^{31} & K_{22}^{31} & K_{23}^{31} & K_{24}^{31} \\ K_{31}^{31} & K_{32}^{31} & K_{33}^{31} & K_{34}^{31} \\ K_{41}^{31} & K_{42}^{31} & K_{43}^{31} & K_{44}^{31} \end{bmatrix} & \begin{bmatrix} K_{11}^{32} & K_{12}^{32} & K_{13}^{32} & K_{14}^{32} \\ K_{21}^{32} & K_{22}^{32} & K_{23}^{32} & K_{24}^{32} \\ K_{31}^{32} & K_{32}^{32} & K_{33}^{32} & K_{34}^{32} \\ K_{41}^{32} & K_{42}^{32} & K_{43}^{32} & K_{44}^{32} \end{bmatrix} & \begin{bmatrix} K_{11}^{33} & K_{12}^{33} & K_{13}^{33} & K_{14}^{33} \\ K_{21}^{33} & K_{22}^{33} & K_{23}^{33} & K_{24}^{33} \\ K_{31}^{33} & K_{32}^{33} & K_{33}^{33} & K_{34}^{33} \\ K_{41}^{33} & K_{42}^{33} & K_{43}^{33} & K_{44}^{33} \end{bmatrix} \end{bmatrix} \tag{3.173}$$

$$\begin{bmatrix} \begin{bmatrix} M_{11}^{11} & M_{12}^{11} & M_{13}^{11} & M_{14}^{11} \\ M_{21}^{11} & M_{22}^{11} & M_{23}^{11} & M_{24}^{11} \\ M_{31}^{11} & M_{32}^{11} & M_{33}^{11} & M_{34}^{11} \\ M_{41}^{11} & M_{42}^{11} & M_{43}^{11} & M_{44}^{11} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} M_{11}^{22} & M_{12}^{22} & M_{13}^{22} & M_{14}^{22} \\ M_{21}^{22} & M_{22}^{22} & M_{23}^{22} & M_{24}^{22} \\ M_{31}^{22} & M_{32}^{22} & M_{33}^{22} & M_{34}^{22} \\ M_{41}^{22} & M_{42}^{22} & M_{43}^{22} & M_{44}^{22} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} M_{11}^{33} & M_{12}^{33} & M_{13}^{33} & M_{14}^{33} \\ M_{21}^{33} & M_{22}^{33} & M_{23}^{33} & M_{24}^{33} \\ M_{31}^{33} & M_{32}^{33} & M_{33}^{33} & M_{34}^{33} \\ M_{41}^{33} & M_{42}^{33} & M_{43}^{33} & M_{44}^{33} \end{bmatrix} \end{bmatrix} \tag{3.174}$$

Rearranging the stiffness and mass matrices according to node where the subscripts represent the local node ID,

$$\begin{bmatrix} \begin{bmatrix} K_{11}^{11} & K_{11}^{12} & K_{11}^{13} \\ K_{11}^{21} & K_{11}^{22} & K_{11}^{23} \\ K_{11}^{31} & K_{11}^{32} & K_{11}^{33} \end{bmatrix} & \begin{bmatrix} K_{12}^{11} & K_{12}^{12} & K_{12}^{13} \\ K_{12}^{21} & K_{12}^{22} & K_{12}^{23} \\ K_{12}^{31} & K_{12}^{32} & K_{12}^{33} \end{bmatrix} & \begin{bmatrix} K_{13}^{11} & K_{13}^{12} & K_{13}^{13} \\ K_{13}^{21} & K_{13}^{22} & K_{13}^{23} \\ K_{13}^{31} & K_{13}^{32} & K_{13}^{33} \end{bmatrix} & \begin{bmatrix} K_{14}^{11} & K_{14}^{12} & K_{14}^{13} \\ K_{14}^{21} & K_{14}^{22} & K_{14}^{23} \\ K_{14}^{31} & K_{14}^{32} & K_{14}^{33} \end{bmatrix} \\ \begin{bmatrix} K_{21}^{11} & K_{21}^{12} & K_{21}^{13} \\ K_{21}^{21} & K_{21}^{22} & K_{21}^{23} \\ K_{21}^{31} & K_{21}^{32} & K_{21}^{33} \end{bmatrix} & \begin{bmatrix} K_{22}^{11} & K_{22}^{12} & K_{22}^{13} \\ K_{22}^{21} & K_{22}^{22} & K_{22}^{23} \\ K_{22}^{31} & K_{22}^{32} & K_{22}^{33} \end{bmatrix} & \begin{bmatrix} K_{23}^{11} & K_{23}^{12} & K_{23}^{13} \\ K_{23}^{21} & K_{23}^{22} & K_{23}^{23} \\ K_{23}^{31} & K_{23}^{32} & K_{23}^{33} \end{bmatrix} & \begin{bmatrix} K_{24}^{11} & K_{24}^{12} & K_{24}^{13} \\ K_{24}^{21} & K_{24}^{22} & K_{24}^{23} \\ K_{24}^{31} & K_{24}^{32} & K_{24}^{33} \end{bmatrix} \\ \begin{bmatrix} K_{31}^{11} & K_{31}^{12} & K_{31}^{13} \\ K_{31}^{21} & K_{31}^{22} & K_{31}^{23} \\ K_{31}^{31} & K_{31}^{32} & K_{31}^{33} \end{bmatrix} & \begin{bmatrix} K_{32}^{11} & K_{32}^{12} & K_{32}^{13} \\ K_{32}^{21} & K_{32}^{22} & K_{32}^{23} \\ K_{32}^{31} & K_{32}^{32} & K_{32}^{33} \end{bmatrix} & \begin{bmatrix} K_{33}^{11} & K_{33}^{12} & K_{33}^{13} \\ K_{33}^{21} & K_{33}^{22} & K_{33}^{23} \\ K_{33}^{31} & K_{33}^{32} & K_{33}^{33} \end{bmatrix} & \begin{bmatrix} K_{34}^{11} & K_{34}^{12} & K_{34}^{13} \\ K_{34}^{21} & K_{34}^{22} & K_{34}^{23} \\ K_{34}^{31} & K_{34}^{32} & K_{34}^{33} \end{bmatrix} \\ \begin{bmatrix} K_{41}^{11} & K_{41}^{12} & K_{41}^{13} \\ K_{41}^{21} & K_{41}^{22} & K_{41}^{23} \\ K_{41}^{31} & K_{41}^{32} & K_{41}^{33} \end{bmatrix} & \begin{bmatrix} K_{42}^{11} & K_{42}^{12} & K_{42}^{13} \\ K_{42}^{21} & K_{42}^{22} & K_{42}^{23} \\ K_{42}^{31} & K_{42}^{32} & K_{42}^{33} \end{bmatrix} & \begin{bmatrix} K_{43}^{11} & K_{43}^{12} & K_{43}^{13} \\ K_{43}^{21} & K_{43}^{22} & K_{43}^{23} \\ K_{43}^{31} & K_{43}^{32} & K_{43}^{33} \end{bmatrix} & \begin{bmatrix} K_{44}^{11} & K_{44}^{12} & K_{44}^{13} \\ K_{44}^{21} & K_{44}^{22} & K_{44}^{23} \\ K_{44}^{31} & K_{44}^{32} & K_{44}^{33} \end{bmatrix} \end{bmatrix} \quad (3.175)$$

$$\begin{bmatrix} \begin{bmatrix} M_{11}^{11} & 0 & 0 \\ 0 & M_{11}^{22} & 0 \\ 0 & 0 & M_{11}^{33} \end{bmatrix} & \begin{bmatrix} M_{12}^{11} & 0 & 0 \\ 0 & M_{12}^{22} & 0 \\ 0 & 0 & M_{12}^{33} \end{bmatrix} & \begin{bmatrix} M_{13}^{11} & 0 & 0 \\ 0 & M_{13}^{22} & 0 \\ 0 & 0 & M_{13}^{33} \end{bmatrix} & \begin{bmatrix} M_{14}^{11} & 0 & 0 \\ 0 & M_{14}^{22} & 0 \\ 0 & 0 & M_{14}^{33} \end{bmatrix} \\ \begin{bmatrix} M_{21}^{11} & 0 & 0 \\ 0 & M_{21}^{22} & 0 \\ 0 & 0 & M_{21}^{33} \end{bmatrix} & \begin{bmatrix} M_{22}^{11} & 0 & 0 \\ 0 & M_{22}^{22} & 0 \\ 0 & 0 & M_{22}^{33} \end{bmatrix} & \begin{bmatrix} M_{23}^{11} & 0 & 0 \\ 0 & M_{23}^{22} & 0 \\ 0 & 0 & M_{23}^{33} \end{bmatrix} & \begin{bmatrix} M_{24}^{11} & 0 & 0 \\ 0 & M_{24}^{22} & 0 \\ 0 & 0 & M_{24}^{33} \end{bmatrix} \\ \begin{bmatrix} M_{31}^{11} & 0 & 0 \\ 0 & M_{31}^{22} & 0 \\ 0 & 0 & M_{31}^{33} \end{bmatrix} & \begin{bmatrix} M_{32}^{11} & 0 & 0 \\ 0 & M_{32}^{22} & 0 \\ 0 & 0 & M_{32}^{33} \end{bmatrix} & \begin{bmatrix} M_{33}^{11} & 0 & 0 \\ 0 & M_{33}^{22} & 0 \\ 0 & 0 & M_{33}^{33} \end{bmatrix} & \begin{bmatrix} M_{34}^{11} & 0 & 0 \\ 0 & M_{34}^{22} & 0 \\ 0 & 0 & M_{34}^{33} \end{bmatrix} \\ \begin{bmatrix} M_{41}^{11} & 0 & 0 \\ 0 & M_{41}^{22} & 0 \\ 0 & 0 & M_{41}^{33} \end{bmatrix} & \begin{bmatrix} M_{42}^{11} & 0 & 0 \\ 0 & M_{42}^{22} & 0 \\ 0 & 0 & M_{42}^{33} \end{bmatrix} & \begin{bmatrix} M_{43}^{11} & 0 & 0 \\ 0 & M_{43}^{22} & 0 \\ 0 & 0 & M_{43}^{33} \end{bmatrix} & \begin{bmatrix} M_{44}^{11} & 0 & 0 \\ 0 & M_{44}^{22} & 0 \\ 0 & 0 & M_{44}^{33} \end{bmatrix} \end{bmatrix} \quad (3.176)$$

These matrices are then assembled into the global matrix. For example, take a mesh made up of four Q4 elements (2x2) as shown in Figure 3.13,

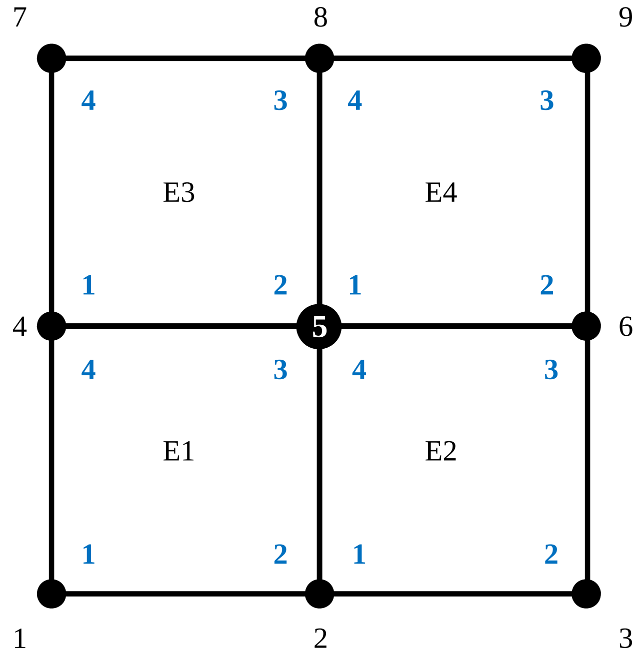



Figure 3.13 Q4 Mesh Example

Using the global ID of the element node rather than the local node ID of the element in the subscripts, the stiffness matrices for the four elements are,

$$[K_{E1}] = \begin{bmatrix} [K_{11}^{lm}] & [K_{12}^{lm}] & [0] & [K_{14}^{lm}] & [K_{15}^{lm}] & [0] & [0] & [0] & [0] \\ [K_{21}^{lm}] & [K_{22}^{lm}] & [0] & [K_{24}^{lm}] & [K_{25}^{lm}] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [K_{41}^{lm}] & [K_{42}^{lm}] & [0] & [K_{44}^{lm}] & [K_{45}^{lm}] & [0] & [0] & [0] & [0] \\ [K_{51}^{lm}] & [K_{52}^{lm}] & [0] & [K_{54}^{lm}] & [K_{55}^{lm}] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \end{bmatrix} \tag{3.177}$$

$$[K_{E2}] = \begin{bmatrix} [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [K_{22}^{lm}] & [K_{23}^{lm}] & [0] & [K_{25}^{lm}] & [K_{26}^{lm}] & [0] & [0] & [0] \\ [0] & [K_{32}^{lm}] & [K_{33}^{lm}] & [0] & [K_{35}^{lm}] & [K_{36}^{lm}] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [K_{52}^{lm}] & [K_{53}^{lm}] & [0] & [K_{55}^{lm}] & [K_{56}^{lm}] & [0] & [0] & [0] \\ [0] & [K_{62}^{lm}] & [K_{63}^{lm}] & [0] & [K_{65}^{lm}] & [K_{66}^{lm}] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \end{bmatrix} \tag{3.178}$$

$$[K_{E3}] = \begin{bmatrix} [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [K_{44}^{lm}] & [K_{45}^{lm}] & [0] & [K_{47}^{lm}] & [K_{48}^{lm}] & [0] \\ [0] & [0] & [0] & [K_{54}^{lm}] & [K_{55}^{lm}] & [0] & [K_{57}^{lm}] & [K_{58}^{lm}] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [K_{74}^{lm}] & [K_{75}^{lm}] & [0] & [K_{77}^{lm}] & [K_{78}^{lm}] & [0] \\ [0] & [0] & [0] & [K_{84}^{lm}] & [K_{85}^{lm}] & [0] & [K_{87}^{lm}] & [K_{88}^{lm}] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \end{bmatrix} \tag{3.179}$$

$$[K_{E4}] = \begin{bmatrix} [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [K_{55}^{lm}] & [K_{56}^{lm}] & [0] & [K_{58}^{lm}] & [K_{59}^{lm}] \\ [0] & [0] & [0] & [0] & [K_{65}^{lm}] & [K_{66}^{lm}] & [0] & [K_{68}^{lm}] & [K_{69}^{lm}] \\ [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [K_{85}^{lm}] & [K_{86}^{lm}] & [0] & [K_{88}^{lm}] & [K_{89}^{lm}] \\ [0] & [0] & [0] & [0] & [K_{95}^{lm}] & [K_{96}^{lm}] & [0] & [K_{98}^{lm}] & [K_{99}^{lm}] \end{bmatrix} \tag{3.180}$$

The nodal matices are abbreviated in the above equations as shown below. The superscripts $l$ and $m$ represent the degrees of freedom in the system.

$$[K_{11}^{lm}] = \begin{bmatrix} K_{11}^{11} & K_{11}^{12} & K_{11}^{13} \\ K_{11}^{21} & K_{11}^{22} & K_{11}^{23} \\ K_{11}^{31} & K_{11}^{32} & K_{11}^{33} \end{bmatrix} \tag{3.181}$$

The matrices are then assembled into the global matrix,

$$[K] = \begin{bmatrix} [K_{11}^{lm}] & [K_{12}^{lm}] & [0] & [K_{14}^{lm}] & [K_{15}^{lm}] & [0] & [0] & [0] & [0] \\ [K_{21}^{lm}] & 2[K_{22}^{lm}] & [K_{23}^{lm}] & [K_{24}^{lm}] & 2[K_{25}^{lm}] & [K_{26}^{lm}] & [0] & [0] & [0] \\ [0] & [K_{32}^{lm}] & [K_{33}^{lm}] & [0] & [K_{35}^{lm}] & [K_{36}^{lm}] & [0] & [0] & [0] \\ [K_{41}^{lm}] & [K_{42}^{lm}] & [0] & 2[K_{44}^{lm}] & 2[K_{45}^{lm}] & [0] & [K_{47}^{lm}] & [K_{48}^{lm}] & [0] \\ [K_{51}^{lm}] & 2[K_{52}^{lm}] & [K_{53}^{lm}] & 2[K_{54}^{lm}] & 4[K_{55}^{lm}] & 2[K_{56}^{lm}] & [K_{57}^{lm}] & 2[K_{58}^{lm}] & [K_{59}^{lm}] \\ [0] & [K_{62}^{lm}] & [K_{63}^{lm}] & [0] & 2[K_{65}^{lm}] & 2[K_{66}^{lm}] & [0] & [K_{68}^{lm}] & [K_{69}^{lm}] \\ [0] & [0] & [0] & [K_{74}^{lm}] & [K_{75}^{lm}] & [0] & [K_{77}^{lm}] & [K_{78}^{lm}] & [0] \\ [0] & [0] & [0] & [K_{84}^{lm}] & 2[K_{85}^{lm}] & [K_{86}^{lm}] & [K_{87}^{lm}] & 2[K_{88}^{lm}] & [K_{89}^{lm}] \\ [0] & [0] & [0] & [0] & [K_{95}^{lm}] & [K_{96}^{lm}] & [0] & [K_{98}^{lm}] & [K_{99}^{lm}] \end{bmatrix} \tag{3.182}$$

For this example, the global stiffness matrix is 27x27. For large systems, storage of these matrices can be expensive. One common method of reducing the size of a symmetric square matrix is to band it into a smaller matrix. The resulting banded global stiffness matrix would be,

$$[K]_{banded} = \begin{bmatrix} [K_{11}^{lm}] & [K_{12}^{lm}] & [0] & [K_{14}^{lm}] & [K_{15}^{lm}] \\ 2[K_{22}^{lm}] & [K_{23}^{lm}] & [K_{24}^{lm}] & 2[K_{25}^{lm}] & [K_{26}^{lm}] \\ [K_{33}^{lm}] & [0] & [K_{35}^{lm}] & [K_{36}^{lm}] & [0] \\ 2[K_{44}^{lm}] & 2[K_{45}^{lm}] & [0] & [K_{47}^{lm}] & [K_{48}^{lm}] \\ 4[K_{55}^{lm}] & 2[K_{56}^{lm}] & [K_{57}^{lm}] & 2[K_{58}^{lm}] & [K_{59}^{lm}] \\ 2[K_{66}^{lm}] & [0] & [K_{68}^{lm}] & [K_{69}^{lm}] & [0] \\ [K_{77}^{lm}] & [K_{78}^{lm}] & [0] & [0] & [0] \\ 2[K_{88}^{lm}] & [K_{89}^{lm}] & [0] & [0] & [0] \\ [K_{99}^{lm}] & [0] & [0] & [0] & [0] \end{bmatrix} \tag{3.183}$$

For this banded matrix, the global stiffness matrix is reduced to a 27x15 matrix, almost half the size of the square matrix. The same process can be applied to the mass matrices. Skyline storage is another scheme that further reduces matrix size. This is only mentioned because it is not the focus of this study.

## 3.5 Finite Element Formulation

The shape functions defined by Cook [29] for a linear four node planar element (Q4) are,

$$\psi_1 = \frac{1}{4}(1-\xi)(1-\eta) \tag{3.184}$$

$$\psi_2 = \frac{1}{4}(1+\xi)(1-\eta) \tag{3.185}$$

$$\psi_3 = \frac{1}{4}(1+\xi)(1+\eta) \tag{3.186}$$

$$\psi_4 = \frac{1}{4}(1-\xi)(1+\eta) \tag{3.187}$$

$$\frac{\partial\psi_1}{\partial\xi} = \frac{1}{4}(\eta-1) \tag{3.188}$$

$$\frac{\partial\psi_1}{\partial\eta} = \frac{1}{4}(\xi-1) \tag{3.189}$$

$$\frac{\partial\psi_2}{\partial\xi} = \frac{1}{4}(1-\eta) \tag{3.190}$$

$$\frac{\partial\psi_2}{\partial\eta} = \frac{1}{4}(-\xi-1) \tag{3.191}$$

$$\frac{\partial\psi_3}{\partial\xi} = \frac{1}{4}(\eta+1) \tag{3.192}$$

$$\frac{\partial\psi_3}{\partial\eta} = \frac{1}{4}(\xi+1) \tag{3.193}$$

$$\frac{\partial\psi_4}{\partial\xi} = \frac{1}{4}(-\eta-1) \tag{3.194}$$

$$\frac{\partial\psi_4}{\partial\eta} = \frac{1}{4}(1-\xi) \tag{3.195}$$

The previously derived stiffness and mass matrices include terms with derivatives of shape function in terms of the global coordinates. The Jacobian is a square matrix that relates derivatives of the shape functions with respect to the element coordinate system, ($\xi$ and $\eta$), to the derivatives of the shape functions with respect to the global coordinate system, ($x$ and $r$).

$$\begin{Bmatrix} \frac{\partial \psi_j}{\partial \xi} \\ \frac{\partial \psi_j}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial r}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial r}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial \psi_j}{\partial x} \\ \frac{\partial \psi_j}{\partial r} \end{Bmatrix} \tag{3.196}$$

The global coordinates are approximated with isoparametric elements as follows,

$$x = \sum_{j=1}^{NPE} x_j \psi_j \tag{3.197}$$

$$r = \sum_{j=1}^{NPE} r_j \psi_j \tag{3.198}$$

$$\frac{dx}{d\xi} = \sum_{j=1}^{NPE} x_j \frac{d\psi_j}{d\xi} \tag{3.199}$$

$$\frac{dx}{d\eta} = \sum_{j=1}^{NPE} x_j \frac{d\psi_j}{d\eta} \tag{3.200}$$

$$\frac{dr}{d\xi} = \sum_{j=1}^{NPE} r_j \frac{d\psi_j}{d\xi} \tag{3.201}$$

$$\frac{dr}{d\eta} = \sum_{j=1}^{NPE} r_j \frac{d\psi_j}{d\eta} \tag{3.202}$$

Substituting (3.199) through (3.202) into (3.196),

$$\begin{Bmatrix} \dfrac{\partial \psi_j}{\partial \xi} \\ \dfrac{\partial \psi_j}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\xi} & \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\xi} \\ \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\eta} & \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\eta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \psi_j}{\partial x} \\ \dfrac{\partial \psi_j}{\partial r} \end{Bmatrix} \tag{3.203}$$

$$[J] = \begin{bmatrix} \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\xi} & \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\xi} \\ \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\eta} & \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\eta} \end{bmatrix} \tag{3.204}$$

For a Q4 element, this may be expanded to,

$$\begin{Bmatrix} \dfrac{\partial \psi_j}{\partial \xi} \\ \dfrac{\partial \psi_j}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \dfrac{d\psi_1}{d\xi} & \dfrac{d\psi_2}{d\xi} & \dfrac{d\psi_3}{d\xi} & \dfrac{d\psi_4}{d\xi} \\ \dfrac{d\psi_1}{d\eta} & \dfrac{d\psi_2}{d\eta} & \dfrac{d\psi_3}{d\eta} & \dfrac{d\psi_4}{d\eta} \end{bmatrix} \begin{bmatrix} x_1 & r_1 \\ x_2 & r_2 \\ x_3 & r_3 \\ x_4 & r_4 \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \psi_j}{\partial x} \\ \dfrac{\partial \psi_j}{\partial r} \end{Bmatrix} \tag{3.205}$$

Equation (3.205) is used to solve for the shape function derivatives with respect to $\xi$ and $\eta$ by inverting the Jacobian, $[J]$,

$$\begin{Bmatrix} \dfrac{\partial \psi_j}{\partial x} \\ \dfrac{\partial \psi_j}{\partial r} \end{Bmatrix} = \begin{bmatrix} \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\xi} & \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\xi} \\ \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\eta} & \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\eta} \end{bmatrix}^{-1} \begin{Bmatrix} \dfrac{\partial \psi_j}{\partial \xi} \\ \dfrac{\partial \psi_j}{\partial \eta} \end{Bmatrix} \tag{3.206}$$

$$\begin{Bmatrix} \dfrac{\partial \psi_j}{\partial x} \\ \dfrac{\partial \psi_j}{\partial r} \end{Bmatrix} = \frac{1}{|J|} \begin{bmatrix} \sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\eta} & -\sum_{j=1}^{NPE} r_j \dfrac{d\psi_j}{d\xi} \\ -\sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\eta} & \sum_{j=1}^{NPE} x_j \dfrac{d\psi_j}{d\xi} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \psi_j}{\partial \xi} \\ \dfrac{\partial \psi_j}{\partial \eta} \end{Bmatrix} \tag{3.207}$$

The scalar, $|J|$, is the determinant of the Jacobian matrix, $[J]$. The inverted Jacobian will be represented by the following notation, $[J^*]$,

$$[J] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{3.208}$$

$$[J]^{-1} = \frac{1}{|J|}\begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \tag{3.209}$$

$$[J^*] = [J]^{-1} \tag{3.210}$$

$$\begin{bmatrix} J_{11}^* & J_{12}^* \\ J_{21}^* & J_{22}^* \end{bmatrix} = \frac{1}{|J|}\begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \tag{3.211}$$

$$\begin{Bmatrix} \dfrac{\partial \psi_j}{\partial x} \\ \dfrac{\partial \psi_j}{\partial r} \end{Bmatrix} = \begin{bmatrix} J_{11}^* & J_{12}^* \\ & \\ J_{21}^* & J_{22}^* \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \psi_j}{\partial \xi} \\ \dfrac{\partial \psi_j}{\partial \eta} \end{Bmatrix} \tag{3.212}$$

Expanding equation (3.212),

$$\frac{\partial \psi_j}{\partial x} = J_{11}^* \frac{\partial \psi_j}{\partial \xi} + J_{12}^* \frac{\partial \psi_j}{\partial \eta} \tag{3.213}$$

$$\frac{\partial \psi_j}{\partial r} = J_{21}^* \frac{\partial \psi_j}{\partial \xi} + J_{22}^* \frac{\partial \psi_j}{\partial \eta} \tag{3.214}$$

Equations (3.213) and (3.214) may now be substituted into equations (3.160) through (3.171). Now all that remains is converting the integrals from $x$ and $r$ coordinates to the natural coordinates, $\xi$ and $\eta$. This is simply achieved by the following relationship,

$$\int_{r_1}^{r_2} \int_{x_1}^{x_2} f(x, r)\, dxdr = \int_{-1}^{1} \int_{-1}^{1} f(\xi, \eta)\, |J|\, d\xi\, d\eta \tag{3.215}$$

Analytical integration of the stiffness and mass terms is computationally expensive. Numerically integrating using Gauss Quadrature is an efficient and accurate approximation and is therefore implemented. This technique evaluates the function at specific sampling points (Gauss points), multiplies the result by a weight factor, and then sums the results over the number of Gauss points. The integrals become,

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\,|J|\,d\xi\,d\eta = \sum_{igp=1}^{NGP}\sum_{jgp=1}^{NGP} f(\xi,\eta)|J|\,W_{igp}\,W_{jgp} \tag{3.216}$$

Where $NGP$ represents the number of Gauss points, $igp$ and $jgp$ represent the Gauss Point of interest, and $W_{igp}$ and $W_{jgp}$ are the weights of the respective Gauss points. Substituting the form of (3.216) into (3.160) through (3.171) completes the finite element formulation. The axisymmetric cylinder with a Q4 mesh shown in Figure 3.14 may now be analyzed.

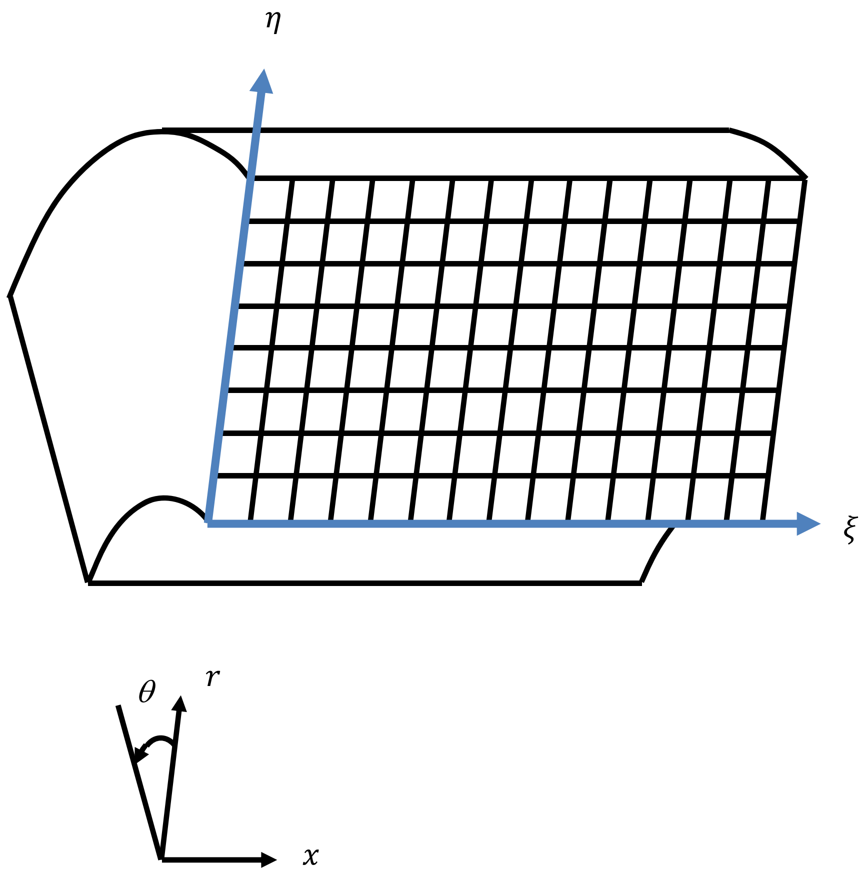



Figure 3.14 Cylinder Coordinates

# CHAPTER 4

## MODAL STRAIN ENERGY METHOD

For the purposes of estimating the dynamic losses in a viscoelastically damped barrel, the Modal Strain Energy Method (MSE) forms the basis of estimating losses due to viscoelastic layers. The Modal Strain Energy Method assumes:

1. The viscoelastic layers are the only source of damping. The fact is ignored that a system will almost always have some sort of damping at interfaces, within materials themselves, and in the medium in which it resides, e.g. air or water. The loss factor of the viscoelastic material is usually at least an order of magnitude greater than most composite materials, Ellis [30] and Oyadiji [31], and even greater than for steels. Therefore, the stainless steel barrel liner and composite material layers are assumed to provide no damping. For this study, the loss factor of the viscoelastic material is assumed to be 1.0 in order to force maximum possible damping effects for the purpose of comparing different barrel layups.
2. The viscoelastic damping was assumed to be independent of temperature and frequency and must be constant. This assumption is obviously false for most analyses but it allows the MSE method to be kept computationally simple. The loss factor and storage modulus for the viscoelastic material are chosen based on an estimated frequency and temperature from the manufacturer's material data sheets. Damping is relative in this analysis. Therefore, the value of the loss factor coefficient is inconsequential to the analysis because only the relative damping in the various layups is being compared.
3. The complex eigenvectors can be approximated with real eigenvectors. This assumption has no theoretical justification. This assumption is made solely to simplify computations. This assumption weakens the theoretical basis of this method because the

damping is contained in the complex eigenvector, and the real eigenvectors are used instead via the strain energies. For large systems, this becomes computational advantageous because a complex eigen solver is no longer required.

4. The strain energy of the structure is calculated using the finite element method. Given appropriate elements to represent the displacement field, the MSE method can accurately calculate the strain energy of a system. The viscoelastic material requires an element that can model three dimensional stress states. Even though this barrel is being analyzed with a two dimensional finite element code, the three degrees of freedom are incorporated into the governing equations and do model the three dimensions.

The MSE makes some very grand assumptions but it is being used in this analysis as a tool to compare relative damping between layups, not the damping in the structure itself. The MSE method has been compared to more refined methods [32] that incorporate temperature and frequency dependencies and has been found to be the most robust and solvable in large systems.

For a discrete finite element model, the equations of motion for the free vibration of a system is be defined as,

$$[M]\{\ddot{x}\} + [K]\{x\} = \{0\} \tag{4.1}$$

Of the many ways that damping may be introduced, a viscous damping model may be chosen for simplicity. The equations of motion become,

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{0\} \tag{4.2}$$

For constant mass, damping, and stiffness matrices, (4.2) is a set of ordinary linear differential equations. However, for viscoelastic materials, the viscous model does not accurately represent damping. The damping in a viscoelastic material is better modeled by using both real and complex components of in the stiffness matrix,

$$[K] = [K]_{\Re} + i[K]_{\Im} \tag{4.3}$$

The imaginary portion of the stiffness is called the loss modulus and it is both frequency and temperature dependent. For simplicity, both the frequency and temperature are assumed to remain constant. The equations of motion become,

$$[M]\{\ddot{x}\} + ([K]_{\Re} + i[K]_{\Im})\{x\} = \{0\} \tag{4.4}$$

Assuming a harmonic solution of,

$$x = U^* e^{i\omega^* t} \tag{4.5}$$

Where,

$$U^* = U_{\Re} + iU_{\Im} \tag{4.6}$$

$$\omega^* = \omega\sqrt{1 + i\eta} \tag{4.7}$$

The loss factor, $\eta$, is now introduced into the damping of the system. Substituting (4.5) into (4.4) yields the complex eigenvalue problem,

$$([K]_{\Re} + i[K]_{\Im})U^* - {\omega^*}^2[M]U^* = [0] \tag{4.8}$$

Multiplying by the transpose of the eigenvector, ${U^*}^T$, the complex eigenvalues may be solved for,

$${\omega^*}^2 = \frac{{U^*}^T([K]_{\Re} + i[K]_{\Im})U^*}{{U^*}^T[M]U^*} \tag{4.9}$$

Substituting (4.7) into (4.9) yields,

$$\omega^2(1 + i\eta) = \frac{{U^*}^T([K]_{\Re} + i[K]_{\Im})U^*}{{U^*}^T[M]U^*} \tag{4.10}$$

Which may be separated into the real and imaginary parts,

$$\omega^2 = \frac{U^{*T}[K]_{\Re}U^*}{U^{*T}[M]U^*} \tag{4.11}$$

$$\omega^2\eta = \frac{U^{*T}[K]_{\Im}U^*}{U^{*T}[M]U^*} \tag{4.12}$$

A damped normal modes analysis would solve equations (4.11) and (4.12) and directly calculate the loss factor, $\eta$. Since this can be very complicated [33], the Modal Strain Energy method makes the assumption that the complex mode shape can be approximated by one computed using a normal modes analysis where damping is ignored (complex eigenvectors are estimated with the real eigenvectors). Hence, equations (4.11) and (4.12) become,

$$\omega^2 = \frac{U^T[K]_{\Re}U}{U^T[M]U} \tag{4.13}$$

$$\omega^2\eta = \frac{U^T[K]_{\Im}U}{U^T[M]U} \tag{4.14}$$

Solving for the loss factor,

$$\eta = \frac{U^T[K]_{\Im}U}{U^T[K]_{\Re}U} \tag{4.15}$$

Noting (4.15) and turning attention towards the laminated cylinder, the stiffness matrix, $[K]$, has contributions from the stainless steel liner, the composite material, and the, viscoelastic layers.

$$[K] = [K]_{SS} + [K]_C + [K]_{VE} \tag{4.16}$$

Representing $[K]_{VE}$ as a complex modulus,

$$[K]_{VE} = [K]_{VE_{\Re}} + i[K]_{VE_{\Im}} \tag{4.17}$$

Substituting (4.17) into (4.16),

$$[K] = \left([K]_{SS} + [K]_C + [K]_{VE_{\Re}}\right) + i[K]_{VE_{\Im}} \tag{4.18}$$

Note that (4.17) may be written as,

$$[K]_{VE} = [K]_{VE_{\Re}}(1 + i\eta_{VE}) \tag{4.19}$$

Hence,

$$[K]_{VE_{\Re}} + i[K]_{VE_{\Im}} = [K]_{VE_{\Re}}(1 + i\eta_{VE}) \tag{4.20}$$

While $\eta_{VE}$ represents the loss factor only in the viscoelastic material and $\eta$ represents the loss factor of the entire barrel. Assuming only a purely elastic normal mode is analyzed, i.e. a real mode ($U = U_{\Re}$), the strain energy for the entire system associated with that mode shape is,

$$V = U^T[K]_{\Re}U \tag{4.21}$$

Where

$$[K]_{\Re} = [K]_{SS} + [K]_C + [K]_{VE_{\Re}} \tag{4.22}$$

Substituting (4.22) into (4.21),

$$V = U^T\left([K]_{SS} + [K]_C + [K]_{VE_{\Re}}\right)U \tag{4.23}$$

Similarly, the portion of the strain energy associated only with the viscoelastic material is,

$$V_{VE} = U^T[K]_{VE_{\Re}}U \tag{4.24}$$

The ratio of the viscoelastic strain energy to the total strain energy of the system is,

$$\frac{V_{VE}}{V} = \frac{U^T[K]_{VE_{\Re}}U}{U^T[K]_{\Re}U} \tag{4.25}$$

Rewriting (4.15) using (4.19) and the fact that the only complex part of $[K]_{\Im}$ occurs only in the viscoelastic layer, i.e. $[K]_{\Im} = [K]_{VE_{\Im}}$,

$$\eta = \frac{U^T[K]_{VE_{\Im}}U}{U^T[K]_{\Re}U} \tag{4.26}$$

With,

$$[K]_{VE_{\Re}} + i[K]_{VE_{\Im}} = [K]_{VE_{\Re}}(1 + i\eta_{VE}) \tag{4.27}$$

$$[K]_{VE_{\Re}} + i[K]_{VE_{\Im}} = [K]_{VE_{\Re}} + i\eta_{VE}[K]_{VE_{\Re}} \tag{4.28}$$

$$i[K]_{VE_{\mathfrak{I}}} = i\eta_{VE}[K]_{VE_{\mathfrak{R}}} \tag{4.29}$$

$$[K]_{VE_{\mathfrak{I}}} = \eta_{VE}[K]_{VE_{\mathfrak{R}}} \tag{4.30}$$

Equation (4.26) becomes,

$$\eta = \frac{U^T \eta_{VE}[K]_{VE_{\mathfrak{R}}} U}{U^T [K]_{\mathfrak{R}} U} \tag{4.31}$$

Rewriting as a ratio of loss factors,

$$\frac{\eta}{\eta_{VE}} = \frac{U^T [K]_{VE_{\mathfrak{R}}} U}{U^T [K]_{\mathfrak{R}} U} \tag{4.32}$$

Equating (4.32) to (4.25) results in,

$$\frac{\eta}{\eta_{VE}} = \frac{V_{VE}}{V} \tag{4.33}$$

Or rewritten as,

$$\eta = \frac{V_{VE}\eta_{VE}}{V} \tag{4.34}$$

Equation (4.34) is the relationship used to determine the loss factor using the Modal Strain Energy method. This means that the strain energy for the total system, the strain energy for the viscoelastic layers, and the loss factor in the viscoelastic layers need to be known in order to determine the loss factor of the total system.

The strain energy density, or the area under the stress strain curve, is defined as,

$$V_o = \int_0^{\varepsilon_{ij}} \sigma_{ij}\, d\varepsilon_{ij} \tag{4.35}$$

Where index notation is used to represent the stress and strain tensors and indicates summation over the indices. The strain energy is calculated by integrating the strain energy density over the volume,

$$V = \int V_o \, d\mathbf{V} \tag{4.36}$$

This is easily incorporated into the finite element formulation and is calculated as part of the static solution.

The modal strain energy method is found to be very accurate for the first modes as Novascone [34] and Ellis [30] have shown. Since only the fundamental radial expansion mode is considered, this method is used with confidence.

# CHAPTER 5

# SOLUTION PROCESS

## 5.1 Static Solution

Consider a cylindrical shell element as shown in Figure 5.1, Timoshenko and Woinowsky-Krieger [35],

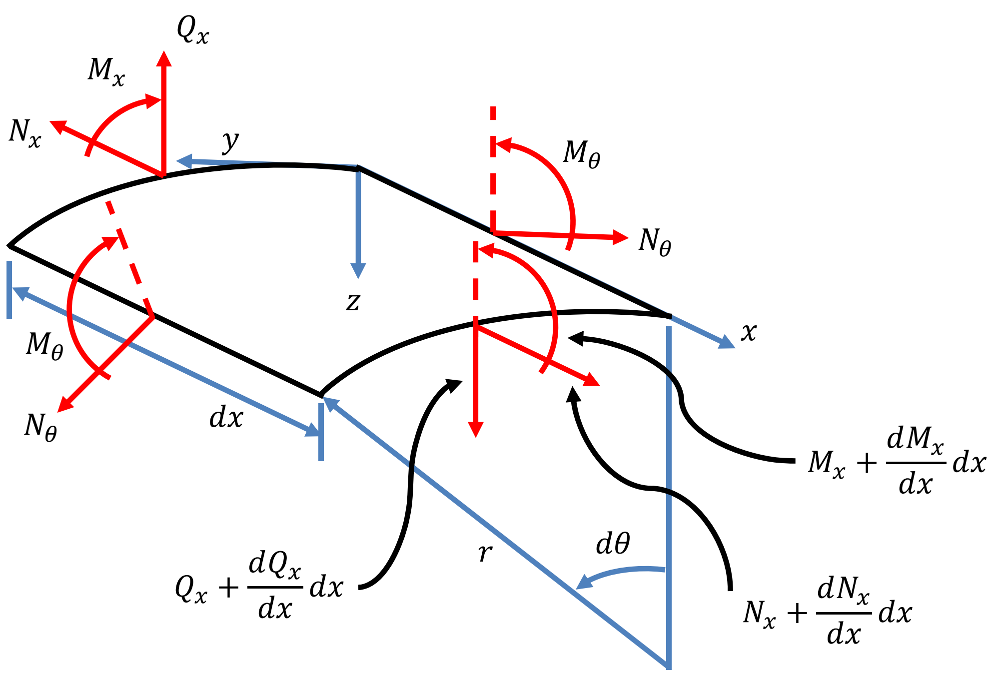



Figure 5.1 Cylindrical Shell Element

By symmetry, $N_{x\theta} = N_{\theta x} = 0$, $N_\theta$ is constant along the circumference, $M_{x\theta} = M_{\theta x} = 0$, $M_\theta$ is constant along the circumference, and only $Q_x$ remains nonzero. Assuming that external forces act normally to the surface, the equations of equilibrium are,

$$\frac{dN_x}{dx} r\, dx\, d\theta = 0 \tag{5.1}$$

$$\frac{dQ_x}{dx} r\,dx\,d\theta + N_\theta\,dx\,d\theta + P_{in} r\,dx\,d\theta = 0 \tag{5.2}$$

$$\frac{dM_x}{dx} r\,dx\,d\theta - Q_x r\,dx\,d\theta = 0 \tag{5.3}$$

Since the circumferential displacement vanishes due to symmetry, only the radial and axial displacements remain and the expressions for strain become,

$$\varepsilon_x = \frac{du_x}{dx} \tag{5.4}$$

$$\varepsilon_\theta = -\frac{u_r}{r} \tag{5.5}$$

Applying Hooke's Law,

$$N_x = \frac{Eh(\varepsilon_x + \nu\varepsilon_\theta)}{1-\nu^2} = 0 \tag{5.6}$$

$$N_\theta = \frac{Eh(\varepsilon_\theta + \nu\varepsilon_x)}{1-\nu^2} = \frac{-Ehu_r}{r} \tag{5.7}$$

For the bending moments, there is no change in curvature in the circumferential direction due to symmetry. The curvatures are,

$$M_x = -\left(\frac{Eh^3}{12(1-\nu^2)}\right)\frac{d^2}{dx^2}u_r \tag{5.8}$$

$$M_\theta = \nu M_x \tag{5.9}$$

Eliminating $Q_x$ from the equations,

$$\frac{d^2}{dx^2}M_x + \frac{N_\theta}{r} = -P_{in} \tag{5.10}$$

Substituting,

$$\frac{d^2}{dx^2}\left(\left(\frac{Eh^3}{12(1-\nu^2)}\right)\frac{d^2}{dx^2}u_r(x)\right) + \frac{Eh}{r^2}u_r(x) = P_{in} \tag{5.11}$$

For constant shell thickness,

$$\left(\frac{Eh^3}{12(1-\nu^2)}\right)\frac{d^4}{dx^4}u_r(x)+\frac{Eh}{r^2}u_r(x)=P_{in} \tag{5.12}$$

Using,

$$\beta^4=\frac{3(1-\nu^2)}{r^2h^2} \tag{5.13}$$

The equation becomes,

$$\frac{d^4}{dx^4}u_r(x)+4\beta^4u_r(x)=\frac{12P_{in}(1-\nu^2)}{Eh^3} \tag{5.14}$$

The general solution to this differential equation, with $C_i$ being unknown constants,

$$u_r(x)=e^{\beta x}(C_1\cos(\beta x)+C_2\sin(\beta x))+e^{-\beta x}(C_3\cos(\beta x)+C_4\sin(\beta x))+f(x) \tag{5.15}$$

The function, $f(x)$, is a particular solution to the differential equation,

$$f(x)=\frac{3P_{in}(\nu^2-1)}{\beta^4Eh^3} \tag{5.16}$$

Since the reaction forces at the end, $x=0$, are balanced and produce local bending which die out rapidly as $x$ increases, $C_1=C_2=0$ by Saint-Venant's Principle. Applying the boundary conditions and solving for $C_3$ and $C_4$,

$$u_r(0)=0 \tag{5.17}$$

$$\left.\frac{d}{dx}u_r(x)\right|_{x=0}=0 \tag{5.18}$$

$$C_3=C_4=\frac{3P_{in}(\nu^2-1)}{\beta^4Eh^3} \tag{5.19}$$

Substituting into the general solution,

$$u_r(x)=\frac{P_{in}r^2}{Eh}\left(1-e^{-\beta x}(C_3\cos(\beta x)-C_4\sin(\beta x))\right) \tag{5.20}$$

As seen in, Figure 5.2, the maximum deflection occurs at,

$$x=\frac{\pi}{\beta} \tag{5.21}$$

$$u_r\left(\frac{\pi}{\beta}\right) = \frac{P_{in}r^2}{Eh}(1 + e^{-\pi}) \tag{5.22}$$

For long cylinders with a constant internal pressure, $P_{in}$, the radial deflection is,

$$u_r(\infty) = \frac{P_{in}r^2}{Eh} \tag{5.23}$$

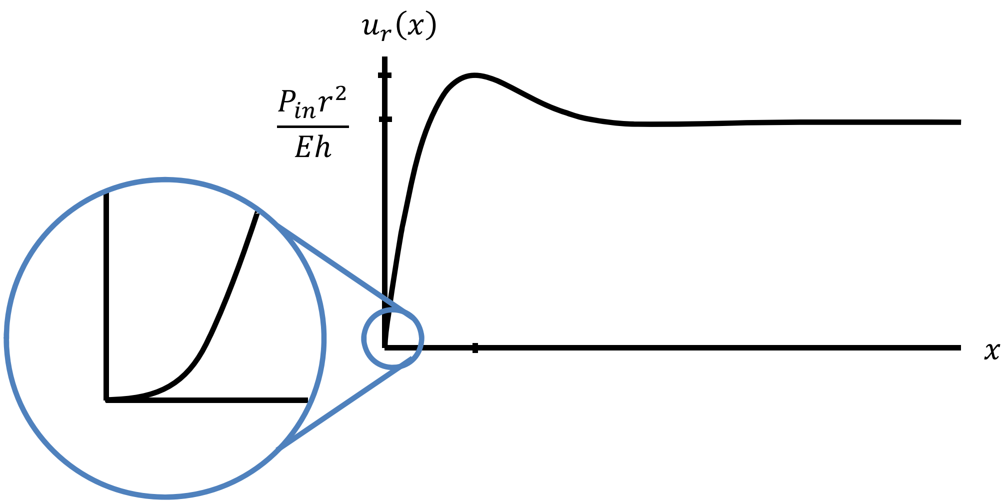



Figure 5.2 Analytical Solution

Although it is not readily apparent in Figure 5.2, the slope at $x = 0$ is equal to zero as required by the boundary condition defined in (5.18). Comparing the analytical solution to the finite element solution, one should not the differences. Shell theory assumes a very small wall thickness. For a gun barrel, the wall thickness is not small but the resulting deflections produced will have the same shape. The finite element model can be compared to the shell model to verify that the model is working correctly as shown in Figure 5.3. Thicker walls, a smaller inside radius, and a coarse mesh account for deviations from the analytical solution. It can be seen that the finite solution also exhibits the same shape as the analytical solution.

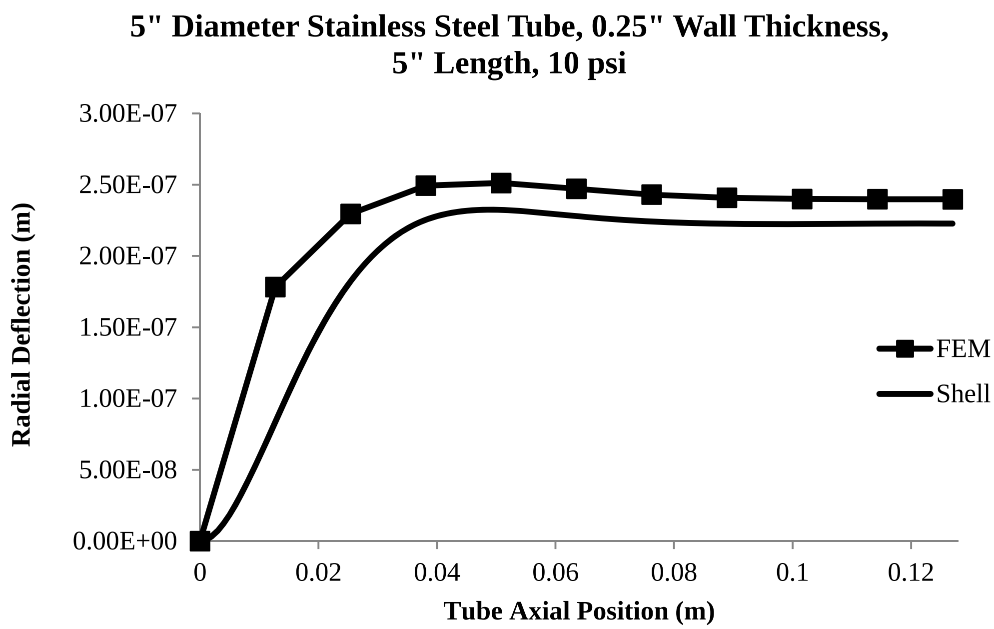



Figure 5.3 FEM vs. Shell Solutions

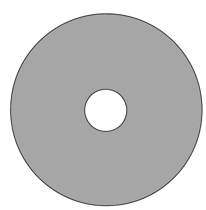
The finite element model solves the static problem, $\{F\} = [K]\{x\}$, by fixing one end of the barrel and using a constant internal pressure applied along the length. Mass is neglected because the system is in equilibrium and there are no inertial forces. The pressure is applied as natural boundary conditions and distributed along the $x$ direction, i.e. forces in the $r$ direction applied at the nodes. The fixed end is enforced through essential boundary conditions by fixing all displacements to zero for all three degrees of freedom. For most guns, this is an accurate assumption because the barrel is usually attached to the firing mechanism at the breach and is not supported along the length. Pressure testing of barrels is accomplished by plugging both the breach and muzzle and then internally pressurizing the system. Deflections are usually measured with a strain gauge.

The barrel being analyzed is a .223 caliber that is 20 inches long and comprised of a stainless steel liner (core tube) that is overwrapped with alternating layers of composite material and viscoelastic material. All interlaminar bonding surfaces are assumed to be perfectly bonded. This could however be modeled using the methods presented by Lyon [36] for various adhesives or even extended to model the effects of Surfi-sculpting. The material properties for the material used in this study are presented in Table 5.1. The dimensions of the barrels analyzed are presented in Table 5.3 through Table 5.8.

Table 5.1 Material Properties

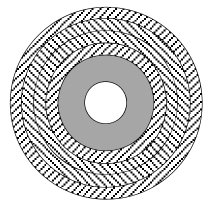
| | $E_1$ (Pa) | $E_2$, $E_3$ (Pa) | $\nu_{23}$ | $\nu_{13}$ | $\nu_{12}$ | $G_{23}$ (Pa) | $G_{13}$, $G_{12}$ (Pa) | $\rho$ (kg/m$^3$) | $\eta$ |
|---|---|---|---|---|---|---|---|---|---|
| Avery 1125 | 132.0E9 | 10.8E9 | 0.59 | 0.24 | | 3.38E9 | 5.65E9 | 1540 | 0 |
| Stainless Steel | 196.5E9 | | 0.27 | | | 77.4E9 | | 2700 | 0 |
| T300/5280 | 2068427.2 | | 0.49 | | | 694103.1 | | 83.18 | 1 |

Table 5.2 Heavy Stainless Steel Barrel



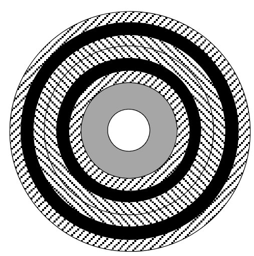
| $R_{in}$ | $R_{out}$ |
|---|---|
| 2.8448 mm | 12.7 mm |

Table 5.3 Barrel #1



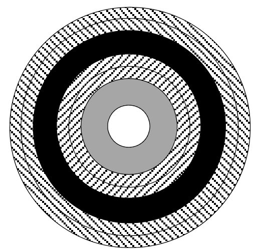
| $R_{in}$ | Radial Thickness (mm) | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|
| | S.S. | +59 | -59 | -59 | +59 | |
| 2.8448 | 3.5052 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 12.7 |

Table 5.4 Barrel #2



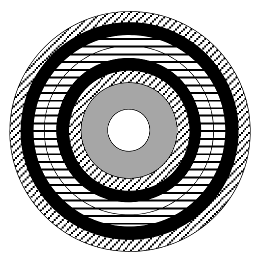
| $R_{in}$ | Radial Thickness (mm) | | | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|---|---|
| | S.S. | +59 | VE | -59 | -59 | VE | +59 | |
| 2.8448 | 3.5052 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 15.875 |

Table 5.5 Barrel #3



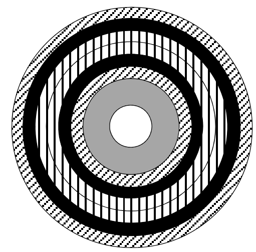
| $R_{in}$ | Radial Thickness (mm) | | | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|---|---|
| | S.S. | +59 | +59 | VE | VE | -59 | -59 | |
| 2.8448 | 3.5052 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 15.875 |

Table 5.6 Barrel #4



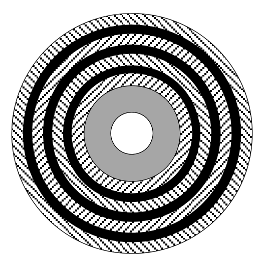
| $R_{in}$ | Radial Thickness (mm) | | | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|---|---|
| | S.S. | +59 | VE | 0 | 0 | VE | -59 | |
| 2.8448 | 3.5052 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 15.875 |

Table 5.7 Barrel #5



| $R_{in}$ | Radial Thickness (mm) | | | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|---|---|
| | S.S. | +59 | VE | 90 | 90 | VE | -59 | |
| 2.8448 | 3.5052 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 15.875 |

Table 5.8 Barrel #6



| $R_{in}$ | Radial Thickness (mm) | | | | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|---|---|---|
| | S.S. | +59 | VE | -59 | VE | +59 | VE | -59 | |
| 2.8448 | 3.5052 | 1.5875 | 1.0583 | 1.5875 | 1.0583 | 1.5875 | 1.0583 | 1.5875 | 15.875 |

This static problem represents the assumed fundamental mode of the barrel as shown in Figure 5.4. With a stainless steel liner providing the majority of the initial stiffness, the interlaminar shear is less than an equivalently stiff composite barrel with no stainless steel liner. The result is a significantly smaller torsional displacement at the free end. The barrel also does not significantly shrink in the axial direction due to the increasing diameter. The static solution was verified for purely composite cylinders with the analytical solution by Herakovich [37].

Figure 5.4 represents the radial displacement of the inside radius relative to the inside radius for the length of the barrel. Note that this is not the radial strain, $\varepsilon_r$.

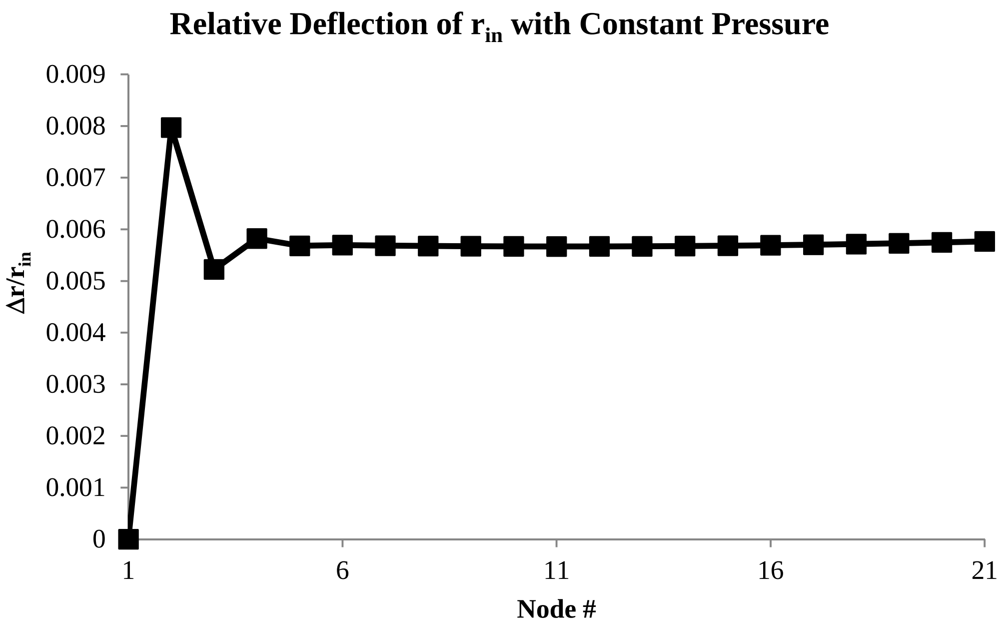



Figure 5.4 Static Displacement, Assumed Fundamental Mode

The loss factors calculated for each of these barrels are presented in Figure 5.5. The angle chosen for the T300/5280 was θ=59 degrees. This is based off the maximum coefficient of mutual influence mismatch between laminae for this material and a purely composite cylinder. This coefficient relates the normal stress and shear stress coupling properties. Figure 5.5 demonstrates the effects of the coefficient of mutual influence and that the alternating off-axis orientation for the layup sequence of Barrel #2 provides a higher loss factor than Barrel #4 and #5. Figure 5.6 illustrates how the off-axis angle increases the loss factor for similar layups. The loss factor for Barrel #6 is larger than Barrel #2, #4, and #5 by a factor of at least 2.6 and is larger than Barrel #3 by a factor of 8.

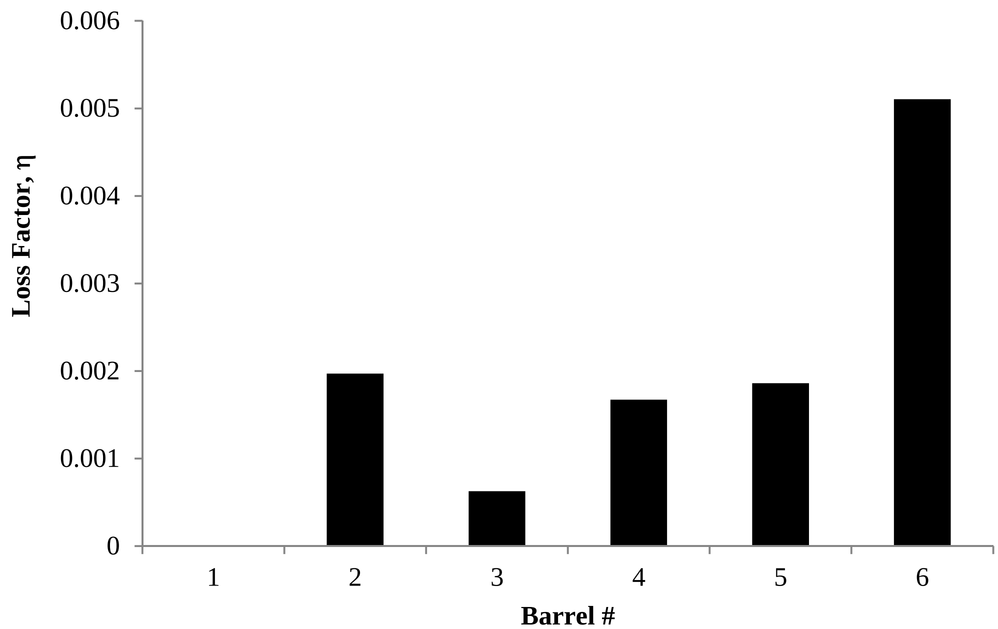



Figure 5.5 Loss Factor Comparison

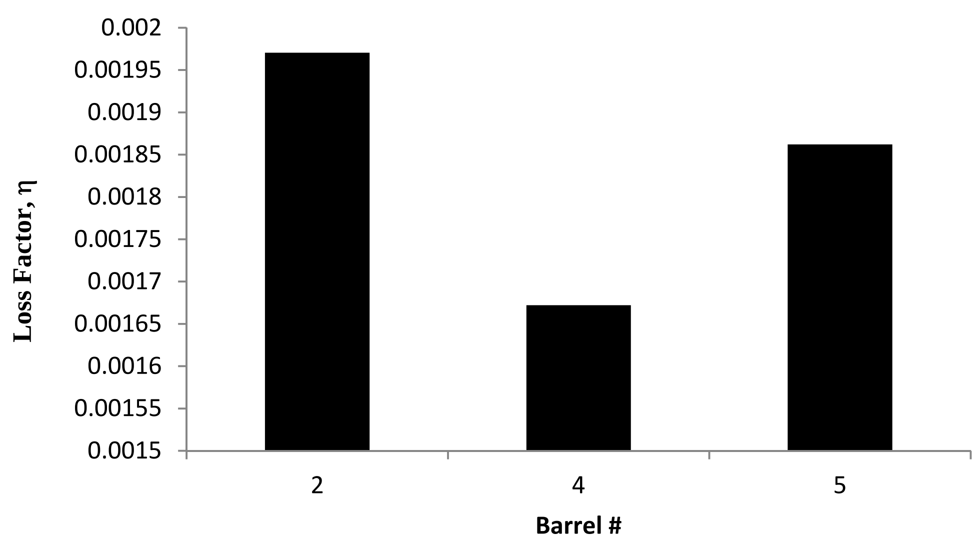



Figure 5.6 Effect of Off-Axis Laminae

Using the layup configuration of Barrel #6 but varying the angle, θ, from 0 to 90 degrees, Figure 5.7 shows how the loss factor changes with the various ply angles. Note that the loss factor spikes for specific angles as seen in Figure 5.8. These spikes are due to the use of a high Poisson's ratio (0.49). Lowering the Poisson's ratio reduces the frequency and magnitudes of the spikes as illustrated in Figure 5.9. For the Barrel #6 configuration, the loss factor peaks at approximately 48.5 degrees after the spikes are filtered out. A new barrel, Barrel #8, may be created with this optimal angle as shown in Table 5.10.

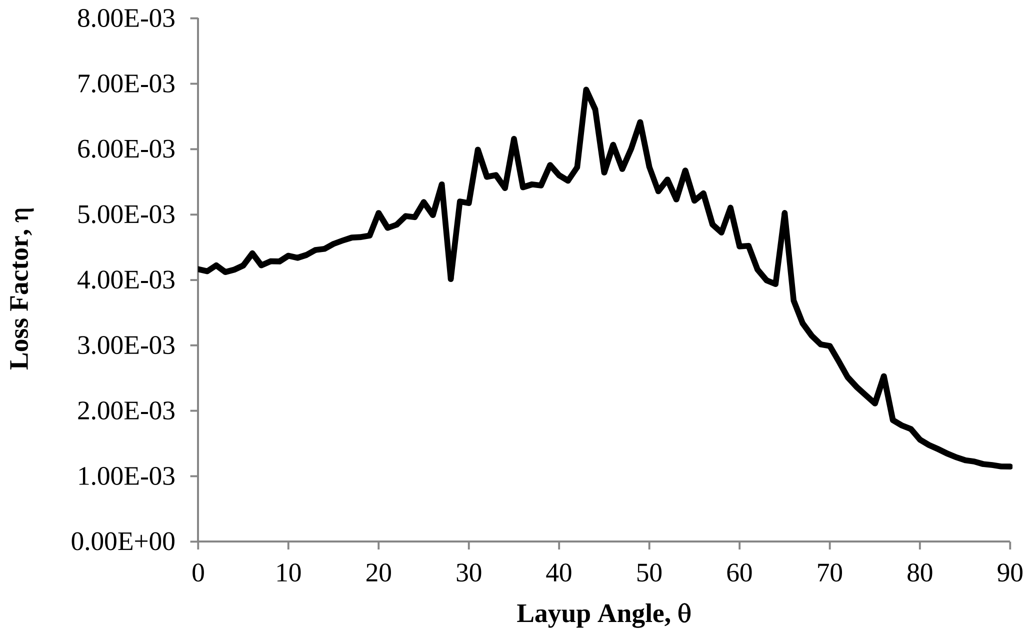



Figure 5.7 Barrel # 8, Varying θ

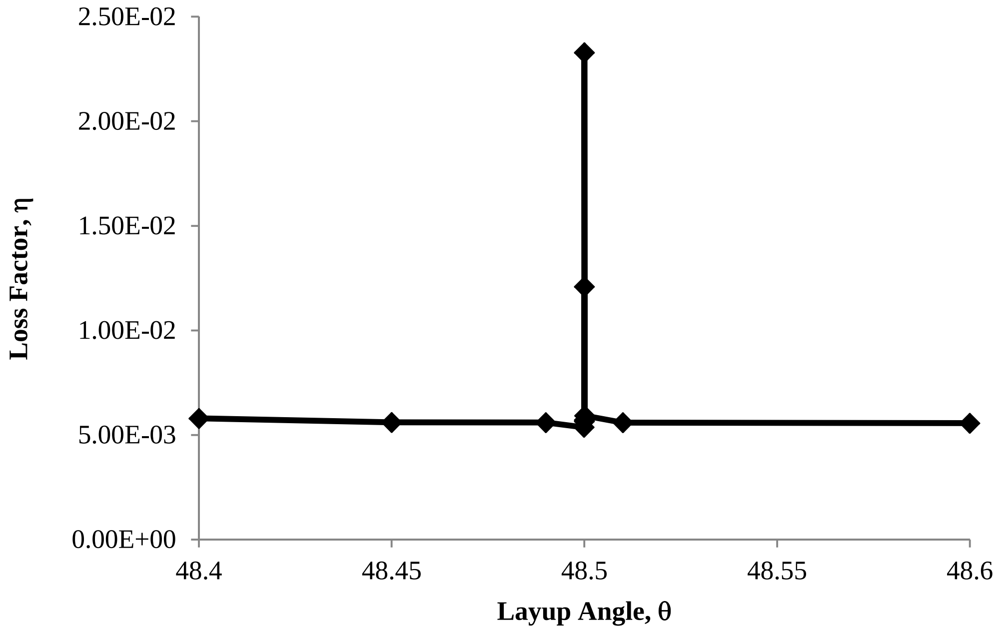



Figure 5.8 Barrel # 6, Varying θ, Spike Detail

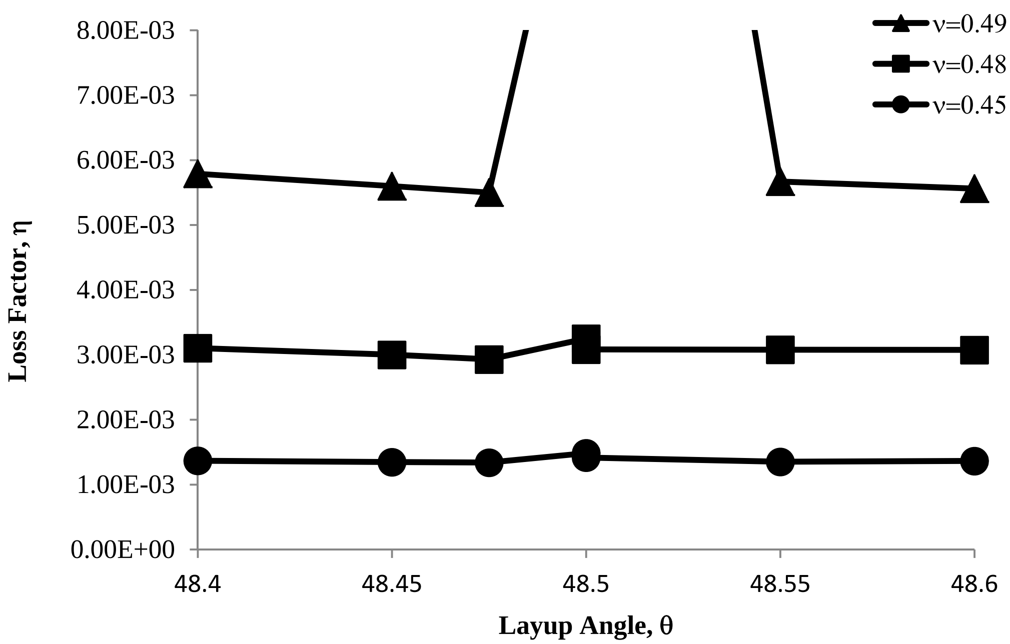
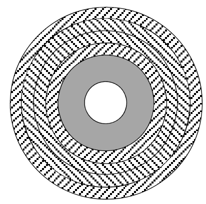


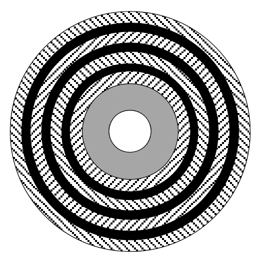

Figure 5.9 Effect of Posson’s Ratio on Loss Factor Spike

Table 5.9 Barrel #7



| $R_{in}$ | Radial Thickness (mm) | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|
| | S.S. | +48.5 | -48.5 | -48.5 | +48.5 | |
| 2.8448 | 3.5052 | 1.5875 | 1.5875 | 1.5875 | 1.5875 | 12.7 |

Table 5.10 Barrel #8



| $R_{in}$ | Radial Thickness (mm) | | | | | | | | $R_{out}$ |
|---|---|---|---|---|---|---|---|---|---|
| | S.S. | +48.5 | VE | -48.5 | VE | +48.5 | VE | -48.5 | |
| 2.8448 | 3.5052 | 1.5875 | 1.0583 | 1.5875 | 1.0583 | 1.5875 | 1.0583 | 1.5875 | 15.875 |

5.2 Dynamic Solution

The dynamic solution is done in two parts. The transient solution of Barrel #1 is obtained (without damping) to find the natural frequency of the system, $\omega_1$. The transient solution of remaining barrels are obtained (without damping) to find the natural frequency of their systems, $\omega_2$. This provides a range of frequencies to begin an analysis. The constant internal pressure is applied along the length of the barrel for the duration of the solution. This is estimated by calculating the average period of oscillations for the length of time of interest (1.5 ms). The analytical solution by Craig [38] for an undamped single degree of freedom problem with the same ideal step input is,

$$u(t) = \frac{p_o}{k}(1 - \cos \omega_n t) \tag{5.24}$$

For the barrel, this is obviously only an estimate for finding the natural frequency without solving an eigenvalue problem. Another possible method for determining frequencies would be a spectral analysis. The time step is important in interpreting results. The transient solution will converge with,

$$\Delta t_{critical} \leq \frac{2}{\omega_{max}} \tag{5.25}$$

The constant average acceleration method is used in the Newmark-Beta scheme, Reddy [39]. This method is unconditionally stable provided equation (5.25) is satisfied. The solution may have different forms if the time step is not sufficiently small to capture the response.

Next, Raleigh damping is used to formulate a damping matrix with the calculated loss factor from the static solution. This damping matrix is purely a mathematical nicety. It has no physical interpretation at all. It merely creates a damping matrix proportional to the mass and stiffness matrices between two given frequencies. At $\omega = \omega_n$, the loss factor may be related by,

$$\eta = 2\zeta \frac{\omega}{\omega_n} \tag{5.26}$$

$$\eta = 2\zeta \tag{5.27}$$

Setting the damping for the two frequencies obtained from the undamped transient solution equal to each other, two equations and two unknowns may be solved to find the Raleigh damping coefficients, $a_0$ and $a_1$,

$$\zeta_1 = \frac{1}{2}\left(\frac{a_0}{\omega_1} + a_1\omega_1\right) \tag{5.28}$$

$$\zeta_2 = \frac{1}{2}\left(\frac{a_0}{\omega_2} + a_1\omega_2\right) \tag{5.29}$$

$$\zeta_1 = \zeta_1 = \zeta = \frac{\eta}{2} \tag{5.30}$$

$$a_0 = \frac{\eta\omega_1\omega_2}{\omega_1 + \omega_2} \tag{5.31}$$

$$a_1 = \frac{\eta}{\omega_1 + \omega_2} \tag{5.32}$$

Using $a_0$ and $a_1$, the viscous damping matrix may now be created,

$$[C] = a_0[M] + a_1[K] \tag{5.33}$$

The last step in producing a dynamic response is running the transient solution a second time with the newly created viscous damping matrix. For this case, the boundary conditions will vary with time as well as along the axial direction. The internal pressure responses are shown in Figure 5.10 and Figure 5.11 below,

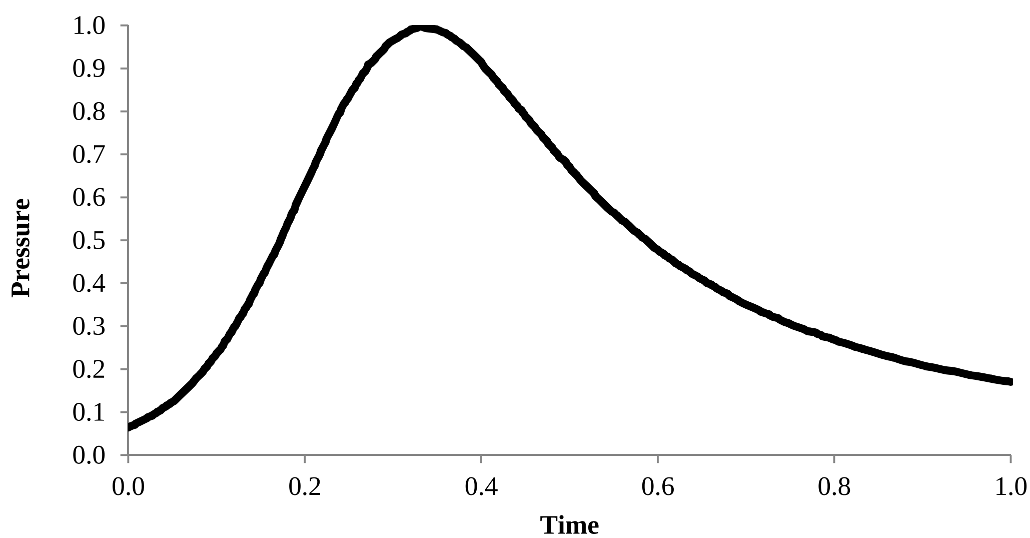



Figure 5.10 Normalized Pressure vs Time

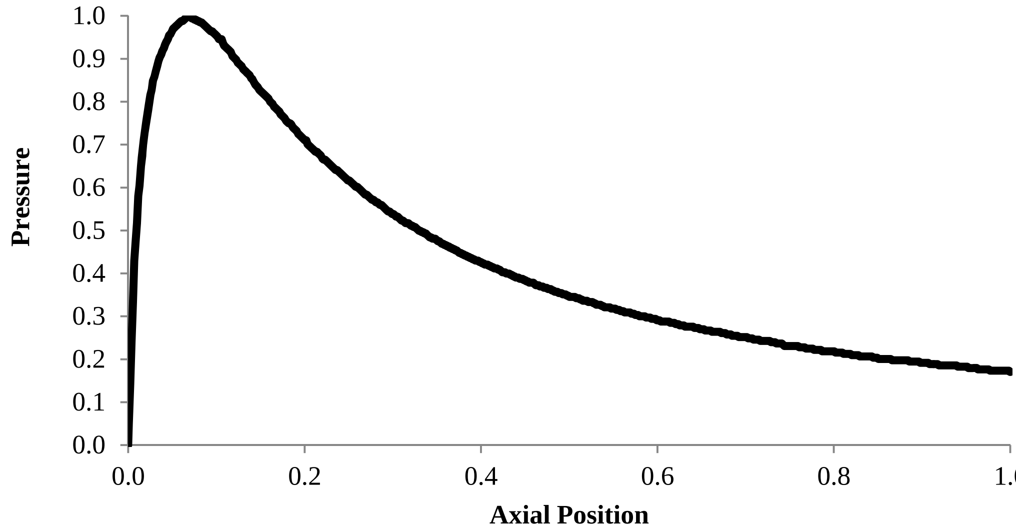



Figure 5.11 Normalized Pressure vs Axial Position

The pressure is multiplied by the max chamber pressure experienced by the gun. For a .223 bullet, that may be as high as 60ksi depending on the charge. The time for the bullet to travel the length of the barrel is used as the max time value. The time step is calculated from the max time and the total number of time steps specified in the input file. From the time in the transient solution, the pressure at that given time is interpolated from the normalized profile in Figure 5.10. From that interpolated pressure, the axial position along the barrel is calculated from Figure 5.11. When the axial position at a specific time is equal to a node position, a natural boundary force is applied at that node with the magnitude specified at that time. The FORTRAN program written for this analysis writes the mass and stiffness arrays from the static solution to a file that is later read in for the dynamic analysis. Inverting large matrices is computationally expensive [40]. This feature speed up solutions by only inverting the matrices for each barrel once. A comparison of Barrel #6 with Barrel #1 is shown below in Figure 5.12.

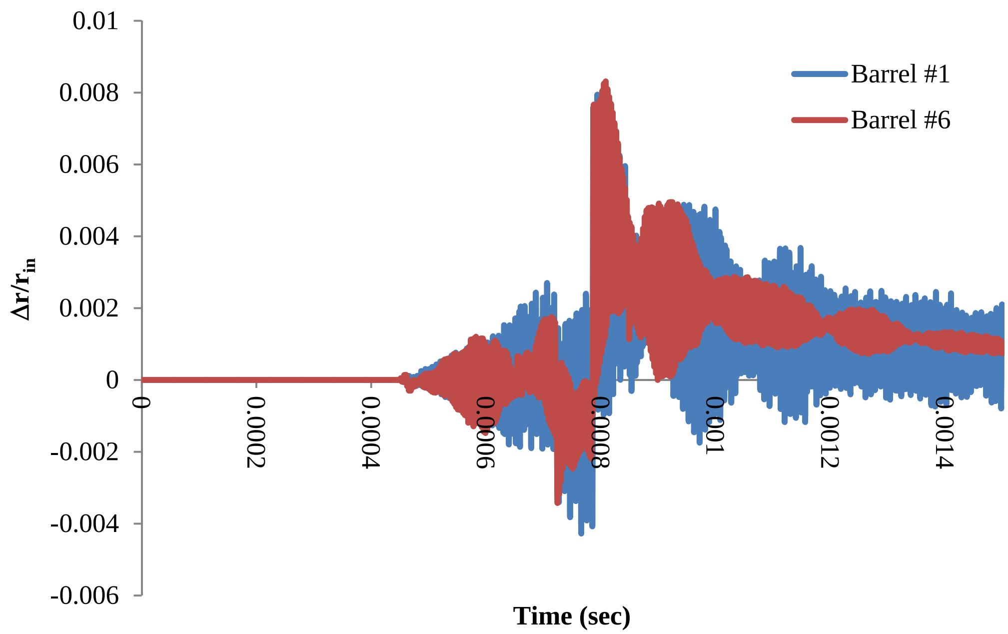



Figure 5.12 Variable Pressure Dynamic Response Comparison

The effects of introducing the damping are noticeable. Interpretation of these results should be taken with care. The intent of this study is not necessarily to model the response of the barrel but to come up with a loss factor and apply it to a model. The model does not necessarily have to be the radial response. Comparing the responses of Barrel #6 and Barrel #8 in Figure 5.14, the response due to the change of angle from 59 degrees to 48.5 degrees while maintaining identical geometry is noticeable after the peak pressure has been applied.

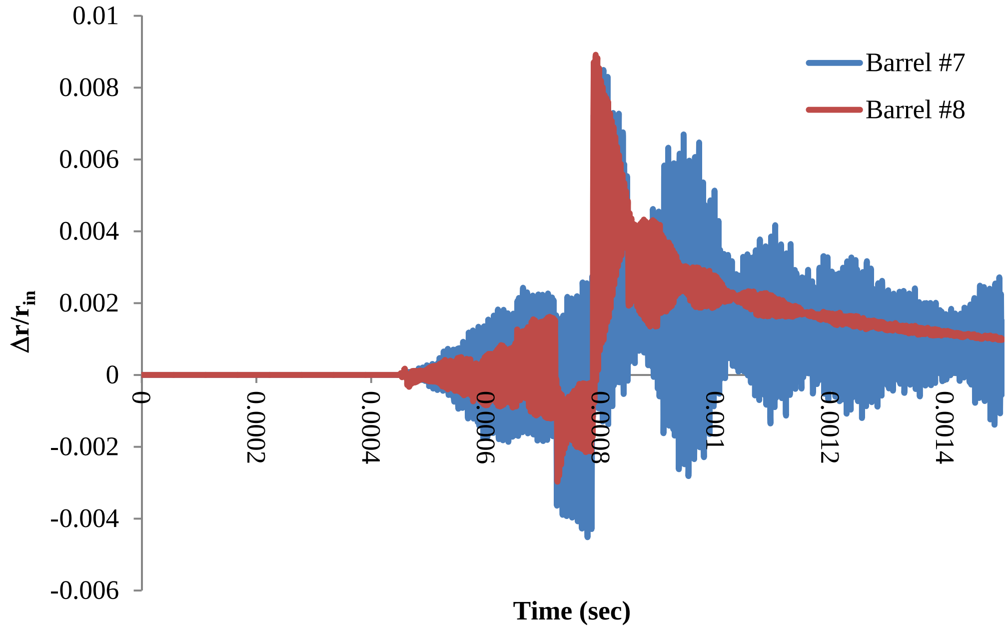



Figure 5.13 Variable Pressure Dynamic Response Comparison, Optimized

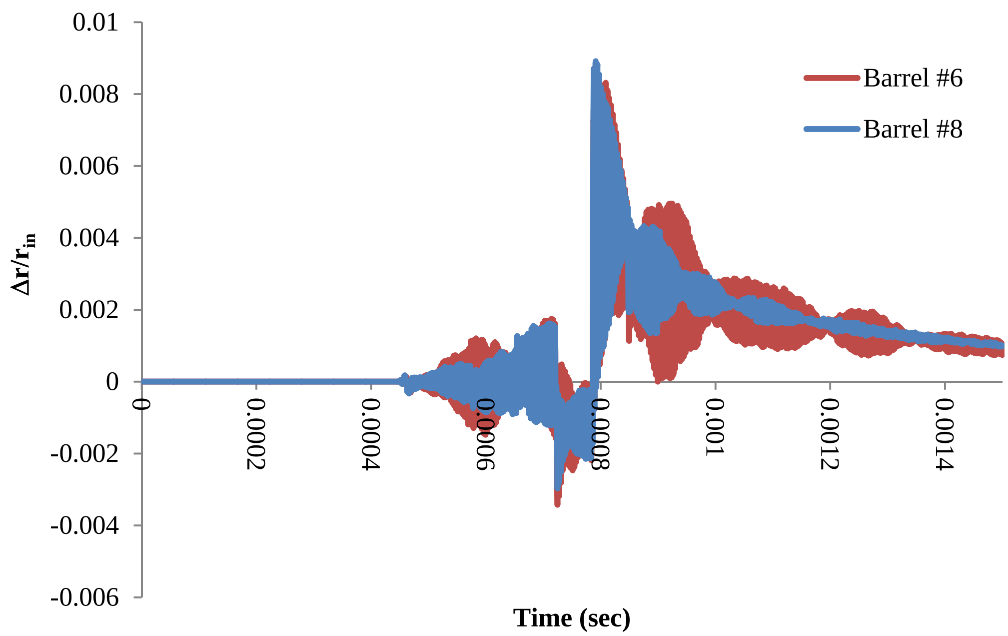



Figure 5.14 Barrel #6 vs. Barrel #8

# CHAPTER 6

# SUMMARY

In summary, the finite element method is used to derive a system of equations to solve a two dimensional axisymmetric model. Results are obtained from the static solution to provide inputs into the dynamic solutions. Using the Modal Strain Energy method, accurate predictions of a loss factor may be generated for a fundamental mode. The material properties of a composite overwrapped barrel may be used to optimize shear stresses between laminae and thus maximize a loss factor. As can be seen with the results of the six barrels, large layers of viscoelastic damping material may not be as effective as thinner layers strategically placed throughout the layup. Furthermore, an optimal angle may be found to maximize the loss factor. The MSE method is a quintessential tool in producing a cheap and accurate comparison of loss factors, which directly translate to the process of optimizing a composite gun barrel with constrained viscoelastic damping layers.

Future advancements in this analysis may include a parametric study on composite material and viscoelastic material used. Several batches of barrels may be manufactured and physically tested to provide a measured comparison to this method. A statistical analysis on the testing may be implemented to measure the significance of the material combinations and provide a confidence interval [41]. The mesh may be optimized by refining it near the fixed boundary rather than across the entire length of the barrel. A study using different axisymmetric elements would also prove to be useful in validating element usage.

# REFERENCES


[1] Sun, C., and Lu, Y., 1995, *Vibration Damping of Structural Elements*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

[2] Musani, A., 2003, "Sound Advice," A10052, from http://www.keepandbeararms.com/information/XcIBViewItem.asp?id=2052.

[3] Department of Defense, 1997, "Design Criteria Standard, Noise Limits," MIL-STD-1474D.

[4] Firehole Consulting Services, 2009, "Analysis of M4-A1 Integral Suppressed Weapon Barrel," Case Study, from http://www.firehole.com/.

[5] Littlefield, A. and Hyland, E., 2006, "Development and Testing of Prestressed Carbon Fiber Composite Overwrapped Gun Tubes," from http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA481065.

[6] Littlefield, A., Hyland, E., Andalora, A., Klein, N., Langone, R. and Becker, R., 2006, "Carbon Fiber/Thermoplastic Overwrapped Gun Tube," J. of Pressure Vessel Technology, **128**, pp. 257-262.

[7] Tzeng, J., 2000, "Dynamic Fracture of Composite Overwrap Cylinders," J. of Reinforced Plastics and Composites, **19** (1), pp. 2-14.

[8] Tzeng, J. and Hopkins, D., 1995, "Dynamic Response of Composite Gun Tubes Subjected to a Moving Internal Pressure," ARL-TR-889, Army Research Laboratory, Aberdeen Proving Ground, MD.

[9] Christensen, R., 1997, "Composite/Metallic Gun Barrel Having a Differing, Restrictive Coefficient of Thermal Expansion," The United States of America, US5657568 A.

[10] Ansari, R., Alisafaei, F. and Ghaedi, P., 2010, "Dynamic Analysis of Multi-layered Filament-wound Composite Pipes Subjected to Cyclic Internal Pressure and Cyclic Temperature," J. of Composite Structures, **92** (5), pp. 1100-1109.

[11] Khalili, S., Malekzadeh, K., Davar, A. and Mahajan, P., "Dynamic Response of Pre-stressed Fiber Metal Laminate (FML) Circular Cylindrical Shells Subjected to Lateral Pressure Pulse Loads," J. of Composite Structures, **92** (6), pp. 1308-1317.

[12] Khalili, S., Azarafza, R. and Davar, A., 2009, "Transient Dynamic Response of Initially Stressed Composite Circular Cylindrical Shells Under Radial Impulse Load," J. of Composite Structures, **89** (2), pp. 275-284.

[13] Setoodeh, A., Tahani, M. and Selahi, E., 2011, "Hybrid Layerwise-Differential Quadrature Transient Dynamic Analysis of Functionally Graded Axisymmetric Cylindrical Shells

Subjected to Dynamic Pressure," J. of Composite Structures, **93** (11), pp. 2663-2670.

[14] Qatu, M., Sullivan, R. and Wang, W., 2010, "Recent Research Advances on the Dynamic Analysis of Composite Shells: 2000–2009," J. of Composite Structures, **93** (1), pp. 14-31.

[15] Qatu, M., 1999, "Theory and Vibration Analysis of Laminated Barrel Thin Shells," J. of Vibration and Control, **5**, pp. 851-889.

[16] Holmberg, B., 1983, "Vibrations of a Taut String and Torsional Vibration of a Gun Barrel Under the Influence of a Moving Mass," J. of Sound and Vibration, **89** (3), pp. 325-334.

[17] Hopkins, D., 1991, "Predicting Dynamic Strain Amplification by Coupling a Finite Element Structural Analysis Code With a Gun Interior Ballistic Code," BRL-TR-3269, Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD.

[18] Robbins, F., Anderson, R. and Gough, P., 1990, "New Pressure Gradient Equations For Lumped-Parameter Interior Ballistic Codes," BRL-TR-3097, Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD.

[19] Chen, M. and South, J., 2007, "Software Development for Automation of Space- and Time-Varying Pressurization on Small Caliber Gun Barrels," ARL-TR-4197, Army Research Laboratory, Aberdeen Proving Ground, MD.

[20] Gonzalez, J., 1990, "Interior Ballistics Optimization," M.S. thesis, Department of Mechanical Engineering, Kansas State University and United States ARMY, AD-A225 791.

[21] Parker, A., Troiano, E. and Underwood, J., 2005, "Stresses Within Compound Tubes Comprising a Steel Liner and an External Carbon-Fiber Wrapped Laminate," J. of Pressure Vessel Technology, **127**, pp. 26-30.

[22] Chen, H., Sun, H. and Liu, T., 2009, "Autofrettage Analysis of a Fibre-reinforced Composite Tube Structure Incorporated with a SMA," J. of Composite Structures, **89** (4), pp. 497-508.

[23] Jense Precision, n.d., from http://www.jenseprecision.com/abs-barrels.html.

[24] Volquartsen, n.d., from https://www.volquartsen.com/tags/5-barrels.

[25] Christensen Arms, n.d., from http://www.christensenarms.com/products/carbon-custom/.

[26] Proof Research, n.d., from http://proofresearch.com/barrels/.

[27] The Welding Institute, n.d., from http://www.twi.co.uk/services/intellectual-property-licensing/surfi-sculpt/.

[28] Hyer, M., 2009, *Stress Analysis of Fiber-Reinforced Composite Materials*, Updated ed.,

DEStech Publications, Inc., Lancaster, PA.

[29] Cook, R., Malkus, D., Plesha, M. and Witt, R., 2001, *Concepts and Applications of Finite Element Analysis*, 4th ed., John Wiley & Sons Inc., Hoboken, NJ.

[30] Ellis, K., 1995, "Damping predictions of Constrained Layers with Orthotropic Face Sheets Using the Modal Strain Energy Method," M.S. thesis, Department of Mechanical Engineering, Utah State University.

[31] Oyadiji, S., 1996, "Characterisation of the Aggregate Complex Modulus Properties of a Fibre-And-Wire-Reinforced Composite Viscoelastic Pipe," *ECCM-7: Seventh European Conference on Composite Materials: Realising Their Commercial Potential*, Institute of Materials, London, pp. 167-172.

[32] Slater, J., Belvin, W. and Inman, D., 1993, "A Survey of Modern Methods for Modeling Frequency Dependent Damping in Finite Element Models," *Proceedings of the 11th Int. Modal Analysis Conference*, **1923**, Society for Experimental Mechanics, Kissimmee, FL, pp. 1508-1512.

[33] Johnson, C. and Kienholz, D., 1982, "Finite Element Prediction of Damping in Structures with Constrained Viscoelastic Layers," *AIAA J.*, **20** (9), pp. 1284-1290.

[34] Novascone, S., 1998, "The Effects of Manufactured Defects on the Axial Damping Characteristics of Composite Specimines," M.S. thesis, Department of Mechanical and Aerospace Engineering, Utah State University.

[35] Timoshenko, S. and Woinowsky-Krieger, S., 1959, *Theory of Plates and Shells*, 2nd ed., McGraw-Hill, Inc., New York, NY.

[36] Lyon, P., 2010, "Axisymmetric Finite Element Modeling for the Design and Analysis of Cylindrical Adhesive Joints Based on Dimensional Stability," M.S. thesis, Department of Mechanical Engineering, Utah State University.

[37] Herakovich, C., 1998, *Mechanics of Fibrous Composites*, John Wiley & Sons, Inc., University of Virginia.

[38] Craig, R. and Kurdila, A., 2006, *Fundamentals of Structural Dynamics*, 2nd ed., John Wiley & Sons Hoboken, NJ.

[39] Reddy, J., 1993, *An Introduction to the Finite Element Method*, 2nd ed. McGraw-Hill, New York, NY.

[40] Chapman, S., 2008, *Fortran 95/2003 for Scientists and Engineers,* 3rd ed., McGraw-Hill, New York, NY.

[41] Coleman, H. and Steele, W., 2009, *Experimentation, Validation, and Uncertainty Analysis for Engineers*, 3rd ed., John Wiley & Sons, Inc., Hoboken, NJ.

# APPENDICES

# Appendix A
FORTRAN Code

```
!------------------------------------------------------------------------------------------------------------------------------!
! STATIC AND DYNAMIC
!------------------------------------------------------------------------------------------------------------------------------!
MODULE VARIABLES

  IMPLICIT NONE   ! Disables default typing provision in FORTRAN. Gives errors for variables not declared. Catches typo's.

  INTEGER::i                                                        ! Do loop variable
  INTEGER::ii                                                       ! Do loop variable
  INTEGER::iii                                                      ! Do loop variable
  INTEGER::iiii                                                     ! Do loop variable
  INTEGER::j                                                        ! Do loop variable
  INTEGER::jj                                                       ! Do loop variable
  INTEGER::jjj                                                      ! Do loop variable
  INTEGER::jjjj                                                     ! Do loop variable
  INTEGER::k                                                        ! Do loop variable
  INTEGER::l                                                        ! Do loop variable
  INTEGER::m                                                        ! Do loop variable
  INTEGER::n                                                        ! Do loop variable

  INTEGER::N_layers                                                 ! N layers

  INTEGER::CASE_FLAG

  REAL(KIND=KIND(0.D0))::x_length                                   ! Length in x direction
  REAL(KIND=KIND(0.D0))::r_length                                   ! Length in x direction
  INTEGER::x_elements                                               ! # of x elements
  INTEGER::r_elements                                               ! # of r elements
  INTEGER::total_elements                                           ! Total # of elements (NEM)
  INTEGER::NPE                                                      ! Nodes Per Element
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::delta_x           ! Evenly spaced distance between nodes in x direction
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::delta_r           ! Evenly spaced distance between nodes in r direction
  INTEGER::x_nodes                                                  ! # of nodes in x direction
  INTEGER::r_nodes                                                  ! # of nodes in r direction
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::x_position        ! x position on mesh
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::r_position        ! r position on mesh
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::t_position        ! theta position on mesh
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::average_x_position        ! x position on mesh
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::average_r_position        ! r position on mesh
  REAL(KIND=KIND(0.D0))::r_eta                                      ! r position in eta coordinates
  INTEGER::total_nodes                                              ! Total # of nodes in mesh (NNM)
  INTEGER,ALLOCATABLE,DIMENSION(:,:)::node                          ! global node number
  INTEGER::NDF                                                      ! Number of degrees of freedom per node
  INTEGER::hbw                                                      ! half band width
  INTEGER::NEQ                                                      ! Total number of equations in the problem
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_stiff           ! GSTIFF
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_mass
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_global_mass1 ! banded matrix multiplication temporary vector
```

```fortran
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_global_mass2
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_global_mass3
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_matrix ! newmark
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix1   ! solve
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix2   ! solve
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix3   ! solve
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix4   ! unband
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix5   ! unband
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix6   ! unband
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_matrix9
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::matrix_inverse   ! inverse
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::inverse_banded
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::mass_inverse
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::stiffness_inverse
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_stiff
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_mass
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_effective_stiff
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_effective_stiffness
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_effective_mass
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_effective_matrix
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::check_inverse
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::square_damp

!REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_vector     ! solve
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_Ka_n   ! newmark
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_Kv_n
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_Ku_n
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_diag
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_matmul1 ! banded matrix multiplication temporary vector
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_matmul2
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_matmul3
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_vector1
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_vector3
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_vector2
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::effective_global_stiffness
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::effective_stiffness
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::effective_global_force
INTEGER,ALLOCATABLE,DIMENSION(:,:)::rep_k_matrix
INTEGER,ALLOCATABLE,DIMENSION(:,:)::rep_m_matrix
INTEGER,ALLOCATABLE,DIMENSION(:,:)::rep_kbar_matrix
INTEGER,ALLOCATABLE,DIMENSION(:,:)::rep_unbanded_matrix
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_force             ! GR
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::element_global_coord   ! ELXY

REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_displacement
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::global_displacement_static
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_velocity
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::global_acceleration
```

```
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_array
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_f
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_d
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_v
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_a
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::temp_m
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_matmul
REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_vector
!REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::temp_array

REAL(KIND=KIND(0.D0)),DIMENSION(3,3)::gauss                           ! Gauss points
REAL(KIND=KIND(0.D0)),DIMENSION(3,3)::weight                          ! Gauss weights
INTEGER::NGP                                                          ! # of Gauss points
REAL(KIND=KIND(0.D0)),DIMENSION(4)::NSF                               ! N, Shape Function
REAL(KIND=KIND(0.D0)),DIMENSION(2,4)::dNSF                            ! dN/dx, derivative of shape function
REAL(KIND=KIND(0.D0))::xi                                             ! x shape function parameter
REAL(KIND=KIND(0.D0))::eta                                            ! y shape function parameter
REAL(KIND=KIND(0.D0))::det                                            ! Determinant of Jacobian
REAL(KIND=KIND(0.D0)),DIMENSION(2,2)::Jac                             ! Jacobian
REAL(KIND=KIND(0.D0)),DIMENSION(2,2)::Jstar                           ! Jacobian Inverse
INTEGER::igp                                                          ! Do loop variable, i Gauss points
INTEGER::jgp                                                          ! Do loop variable, j Gauss points
REAL(KIND=KIND(0.D0)),DIMENSION(2,4)::gdSF                            ! global derivative of shape function

REAL(KIND=KIND(0.D0))::P_constant
REAL(KIND=KIND(0.D0))::P_0

REAL(KIND=KIND(0.D0)),DIMENSION(12)::ELF
REAL(KIND=KIND(0.D0)),DIMENSION(3,3,4,4)::TK
REAL(KIND=KIND(0.D0)),DIMENSION(4,4,3,3)::ELKT
REAL(KIND=KIND(0.D0)),DIMENSION(16,16)::ELK                           ! ndf*npe,ndf*npe
REAL(KIND=KIND(0.D0)),DIMENSION(16)::R                                ! ndf*npe
REAL(KIND=KIND(0.D0)),DIMENSION(3,3,4,4)::TM
REAL(KIND=KIND(0.D0)),DIMENSION(4,4,3,3)::ELMT
REAL(KIND=KIND(0.D0)),DIMENSION(16,16)::ELM                           ! ndf*npe,ndf*npe
REAL(KIND=KIND(0.D0)),DIMENSION(16,16)::ELEFFECTIVE

INTEGER::nw                                                           ! bandwidth

REAL(KIND=KIND(0.D0))::yc
INTEGER::nj
INTEGER::nn                                                           ! NDF*NPE
INTEGER::ni
INTEGER::col
INTEGER::lcol
INTEGER::colbase
INTEGER::rowbase
INTEGER::jdof
```

```
  INTEGER::row
  INTEGER::lrow
  INTEGER::idof

  REAL(KIND=KIND(0.D0))::dpsi_i_dx
  REAL(KIND=KIND(0.D0))::dpsi_j_dx
  REAL(KIND=KIND(0.D0))::dpsi_i_dr
  REAL(KIND=KIND(0.D0))::dpsi_j_dr

  INTEGER::nbdy                                                 ! # of essential Boundary conditions
  INTEGER::nbf                                                  ! # of natural boundary conditions (boundary forces)

  INTEGER::id
  INTEGER::aob
  INTEGER::bwlimit
!  INTEGER::NEQ       ! total number of degrees of freedom in mesh (NDF*NNM)
  REAL(KIND=KIND(0.D0))::val

  INTEGER::nb
  INTEGER,ALLOCATABLE,DIMENSION(:)::ibdy                        ! location of ith constrained EBC
  INTEGER,ALLOCATABLE,DIMENSION(:)::nodbdy                      ! global node id of ith EBC
  INTEGER,ALLOCATABLE,DIMENSION(:)::dofbdy                      ! degree of freedom index of ith EBC
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::vbdy          ! value of ith EBC constraint
  INTEGER,ALLOCATABLE,DIMENSION(:)::ibf                         ! location of ith constrained NBC
  INTEGER,ALLOCATABLE,DIMENSION(:)::nodbf                       ! global node id of ith NBC
  INTEGER,ALLOCATABLE,DIMENSION(:)::dofbf                       ! degree of freedom index of ith NBC
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::vbf           ! value of ith NBC constraint

  INTEGER::ncol,nrow,npiv,npivot,meqns,icol,jcol,ijk,lstsub,jki
  REAL(KIND=KIND(0.D0))::factor

  REAL(KIND=KIND(0.D0))::CB11                           ! Stiffness matrix value, C11
  REAL(KIND=KIND(0.D0))::CB12                           ! Stiffness matrix value, C12
  REAL(KIND=KIND(0.D0))::CB13                           ! Stiffness matrix value, C13
  REAL(KIND=KIND(0.D0))::CB16                           ! Stiffness matrix value, C16
  REAL(KIND=KIND(0.D0))::CB22                           ! Stiffness matrix value, C22
  REAL(KIND=KIND(0.D0))::CB23                           ! Stiffness matrix value, C23
  REAL(KIND=KIND(0.D0))::CB26                           ! Stiffness matrix value, C26
  REAL(KIND=KIND(0.D0))::CB33                           ! Stiffness matrix value, C33
  REAL(KIND=KIND(0.D0))::CB36                           ! Stiffness matrix value, C36
  REAL(KIND=KIND(0.D0))::CB44                           ! Stiffness matrix value, C44
  REAL(KIND=KIND(0.D0))::CB45                           ! Stiffness matrix value, C45
  REAL(KIND=KIND(0.D0))::CB55                           ! Stiffness matrix value, C55
  REAL(KIND=KIND(0.D0))::CB66                           ! Stiffness matrix value, C66

  REAL(KIND=KIND(0.D0))::pi

  REAL(KIND=KIND(0.D0))::P_temp
```

```
  INTEGER::CASE_STATUS                                     ! Flag for selecting case
  INTEGER::SYMMETRIC_LAYUP                                 ! Flag for creating a symmetric lamina
  INTEGER::EQUAL_MAT_PROP                                  ! Flag for selecting equal material properties
  INTEGER,ALLOCATABLE,DIMENSION(:)::materialprop        ! Material property of Nth layer
  INTEGER,ALLOCATABLE,DIMENSION(:)::r_elements_per_layer
  INTEGER,ALLOCATABLE,DIMENSION(:)::total_elements_per_layer
  INTEGER,ALLOCATABLE,DIMENSION(:)::layer_temp
  INTEGER::MATERIAL_FLAG

  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::thetak         ! Ply angle of Nth layer
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::z             ! z location
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::hk             ! Ply thickness of Nth layer
  REAL(KIND=KIND(0.D0))::equal_layer_thickness                      ! Equal layer thickness
  REAL(KIND=KIND(0.D0))::E1                                         ! Modulus in the 1 direction
  REAL(KIND=KIND(0.D0))::E2                                         ! Modulus in the 2 direction
  REAL(KIND=KIND(0.D0))::E3                                         ! Modulus in the 3 direction
  REAL(KIND=KIND(0.D0))::v23             ! Poisson's Ratio relating contraction in the 3 direction to extension in the 2 direction
  REAL(KIND=KIND(0.D0))::v13             ! Poisson's Ratio relating contraction in the 3 direction to extension in the 1 direction
  REAL(KIND=KIND(0.D0))::v12             ! Poisson's Ratio relating contraction in the 2 direction to extension in the 1 direction
  REAL(KIND=KIND(0.D0))::G23                                        ! Shear Modulus in the 2-3 plane
  REAL(KIND=KIND(0.D0))::G13                                        ! Shear Modulus in the 1-3 plane
  REAL(KIND=KIND(0.D0))::G12                                        ! Shear Modulus in the 1-2 plane
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::rho
  REAL(KIND=KIND(0.D0))::rhok
  REAL(KIND=KIND(0.D0))::density
  REAL(KIND=KIND(0.D0))::H                                          ! Total Thickness of Laminate, H
  REAL(KIND=KIND(0.D0))::COS_theta                                  ! Cosine(theta)
  REAL(KIND=KIND(0.D0))::SIN_theta                                  ! Sine(theta)
  REAL(KIND=KIND(0.D0))::v32             ! Poisson's Ratio relating contraction in the 2 direction to extension in the 3 direction
  REAL(KIND=KIND(0.D0))::v31             ! Poisson's Ratio relating contraction in the 1 direction to extension in the 3 direction
  REAL(KIND=KIND(0.D0))::v21                                        ! Conjugate Poisson's Ratio, v21
  REAL(KIND=KIND(0.D0))::v                                          ! C calculation variable
  REAL(KIND=KIND(0.D0))::C11                                        ! Stiffness matrix value, C11
  REAL(KIND=KIND(0.D0))::C12                                        ! Stiffness matrix value, C12
  REAL(KIND=KIND(0.D0))::C13                                        ! Stiffness matrix value, C13
  REAL(KIND=KIND(0.D0))::C22                                        ! Stiffness matrix value, C22
  REAL(KIND=KIND(0.D0))::C23                                        ! Stiffness matrix value, C23
  REAL(KIND=KIND(0.D0))::C33                                        ! Stiffness matrix value, C33
  REAL(KIND=KIND(0.D0))::C44                                        ! Stiffness matrix value, C44
  REAL(KIND=KIND(0.D0))::C55                                        ! Stiffness matrix value, C55
  REAL(KIND=KIND(0.D0))::C66                                        ! Stiffness matrix value, C66
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:)::Cbar          ! Off-axis Stiffness matrix value, C11
  REAL(KIND=KIND(0.D0))::deg_to_rad                                ! Convert degrees to radians for use in intrinsic trig functions
  REAL(KIND=KIND(0.D0))::rad_to_deg
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::r_layer_loc       ! radial locations of each layer
!  REAL(KIND=KIND(0.D0))::length                                    ! cylinder length
```

```
  REAL(KIND=KIND(0.D0))::lr_factor                                  ! Length to inside radius factor

  INTEGER::layer                                                    ! temporary variable to input layer properties for Cbar
  INTEGER::layer2
  INTEGER,ALLOCATABLE,DIMENSION(:)::layer_ID

  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::x_displacement
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::r_displacement
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::t_displacement
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::x_displacement_static
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::r_displacement_static
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::t_displacement_static
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::epsilon_x
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::epsilon_r
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::epsilon_t
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::gamma_tr
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::gamma_rx
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::gamma_xt
!$$$$$$
!$$$$$$   REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::assumed_mode_shape_x
!$$$$$$   REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::assumed_mode_shape_r
!$$$$$$   REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::assumed_mode_shape_t
!$$$$$$   REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::assumed_mode_shape_all
  REAL(KIND=KIND(0.D0))::loss_factor ! kelly ellis eqn 3.27 result
  REAL(KIND=KIND(0.D0))::visco_loss_factor ! material property
  REAL(KIND=KIND(0.D0))::visco_loss_factor2
  REAL(KIND=KIND(0.D0))::strain_energy
  REAL(KIND=KIND(0.D0))::strain_energy_visco
  REAL(KIND=KIND(0.D0))::omega1
  REAL(KIND=KIND(0.D0))::omega2

  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::shared_elements
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::strain


  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:,:)::sigma_g_p
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:,:)::epsilon_g_p

  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:)::r_g_p
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:)::x_g_p
  REAL(KIND=KIND(0.D0))::u_x_g_p                              ! u=x
  REAL(KIND=KIND(0.D0))::u_theta_g_p                          ! v=theta
  REAL(KIND=KIND(0.D0))::u_r_g_p                              ! w=r
!  REAL(KIND=KIND(0.D0))::r_eta
  REAL(KIND=KIND(0.D0))::x_xi
  REAL(KIND=KIND(0.D0))::du_x_dx                              ! d(u_x)/dx
  REAL(KIND=KIND(0.D0))::du_x_dr                              ! d(u_x)/dr
  REAL(KIND=KIND(0.D0))::du_theta_dx                          ! d(u_theta)/dx
```

```
  REAL(KIND=KIND(0.D0))::du_theta_dr                              ! d(u_theta)/dr
  REAL(KIND=KIND(0.D0))::du_r_dx                                  ! d(u_r)/dx
  REAL(KIND=KIND(0.D0))::du_r_dr                                  ! d(u_r)/dr
  REAL(KIND=KIND(0.D0))::root_3
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:)::nodal_strain
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:,:)::nodal_stress
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::average_nodal_stress
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::average_nodal_strain



  REAL(KIND=KIND(0.D0))::P_max
  REAL(KIND=KIND(0.D0))::V_max
  REAL(KIND=KIND(0.D0))::time
  REAL(KIND=KIND(0.D0))::time_post_muzzle
  REAL(KIND=KIND(0.D0))::post_muzzle_factor
  REAL(KIND=KIND(0.D0))::delta_time
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::t,p_interpolated,v_interpolated,x_interpolated
!$$$$$$   REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::p_interpolated_x
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:,:)::P_variable
  INTEGER::n_timesteps
  INTEGER::tt,ttt
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::normal_time_v_vs_t,normal_time_p_vs_t,normal_position_v_vs_x
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::normal_position_p_vs_x
  REAL(KIND=KIND(0.D0)),ALLOCATABLE,DIMENSION(:)::normal_v_vs_t,normal_p_vs_t,normal_v_vs_x,normal_p_vs_x
  REAL(KIND=KIND(0.D0))::x1,x2,y1,y2
!$$$$$$   REAL(KIND=KIND(0.D0))::bullet_mass
  !REAL(KIND=KIND(0.D0))::bullet_velocity
!$$$$$$   REAL(KIND=KIND(0.D0))::normal_max
  INTEGER::interpolated_max
  REAL(KIND=KIND(0.D0))::dt
 ! REAL(KIND=KIND(0.D0))::a0
  !REAL(KIND=KIND(0.D0))::a1
  INTEGER::counter
  REAL(KIND=KIND(0.D0))::beta
!$$$$$$   REAL(KIND=KIND(0.D0))::alpha
  REAL(KIND=KIND(0.D0))::gamma

  real(kind=kind(0.d0)),ALLOCATABLE,dimension(:,:):: aa,cc,LL,UU
  real(kind=kind(0.d0)),ALLOCATABLE,dimension(:)::bb, dd, xx
  real(kind=kind(0.d0)):: coeff


!$$$$$$   real(kind=kind(0.d0))::a0,a1,a2,a3,a4,a5,a6,a7

  INTEGER,PARAMETER::DBL=KIND(0.D0)

  REAL(KIND=KIND(0.D0))::T1,T2,T3,T4,T5,T6,T7,T8
```

```
END MODULE VARIABLES
!-------------------------------------------------------------------------------------------------------------------------------!


!-------------------------------------------------------------------------------------------------------------------------------!
PROGRAM THESIS_FINITE_ELEMENT

  USE VARIABLES
  IMPLICIT NONE

  CALL CPU_TIME(T1)

  pi=3.14159265358979323846264338327 9D0

  WRITE(*,*) "BEGIN PROGRAM"
  WRITE(*,*) "Double Preceision KIND",DBL


  OPEN ( 6, FILE = 'THESIS_FE_4_input.TXT',STATUS = 'unknown')            ! Open file for input
  !OPEN ( 8, FILE = 'THESIS_FE_4_global_stiffness.TXT',STATUS = 'unknown') ! Open file for output
  OPEN (9, FILE = 'THESIS_FINITE_ELEMENT_stress_strain_output.CSV',STATUS = 'unknown')       ! Open file for output
  OPEN (10, FILE = 'THESIS_FE_4_Dynamic_output.CSV',STATUS = 'unknown')      ! Open file for output
  !OPEN (11, FILE = 'THESIS_FE_4_matrices.CSV',STATUS = 'unknown')         ! Open file for output
  OPEN (20, FILE = 'THESIS_FE_4_Static_output.CSV',STATUS = 'unknown')
  OPEN (30, FILE = 'THESIS_FE_4_interpolated_values.CSV',STATUS = 'unknown')   ! Open file for output
  OPEN (40, FILE = 'THESIS_FE_4_lossfactor.TXT',STATUS = 'unknown')   ! Open file for output
  OPEN (50, FILE = 'THESIS_FE_4_mass_inverse.TXT',STATUS = 'unknown')   ! Open file for output
  OPEN (60, FILE = 'THESIS_FE_4_eff_inverse.TXT',STATUS = 'unknown')   ! Open file for output

  deg_to_rad=pi/180.D0
  rad_to_deg=180.D0/pi

  CALL GEOMETRY        ! Calculate z locations
  CALL C_BAR
  tt=0                 ! Calculate C matrices
  CALL NUMBERING
  CALL STEP_0
  CALL CPU_TIME(T2)
  CALL WRITE_STUFF     ! line 1725



END PROGRAM THESIS_FINITE_ELEMENT
!-------------------------------------------------------------------------------------------------------------------------------!


!-------------------------------------------------------------------------------------------------------------------------------!
```

```
SUBROUTINE GEOMETRY

  USE VARIABLES
  IMPLICIT NONE

  !WRITE(*,*) " + BEGIN SUBROUTINE GEOMETRY ( 1/10)"

  READ(6,*) CASE_FLAG ! 1=STATIC, 2=DYNAMIC

  READ(6,*) N_layers                           ! N layers

  ALLOCATE(hk(N_layers))
  ALLOCATE(z(0:N_layers))
  ALLOCATE(thetak(N_layers))
  ALLOCATE(materialprop(N_layers))
  ALLOCATE(r_elements_per_layer(N_layers))
  ALLOCATE(total_elements_per_layer(N_layers))
  ALLOCATE(layer_temp(N_layers))
  ALLOCATE(rho(N_layers))
  ALLOCATE(Cbar(6,6,N_layers))
  ALLOCATE(r_layer_loc(0:N_layers))

  DO i=1,N_layers
    READ(6,*) hk(i)
  END DO

  DO i=1,N_layers
    READ(6,*) thetak(i)
  END DO

  DO i=1,N_layers
    READ(6,*) materialprop(i)
  END DO

  DO i=1,N_layers
    READ(6,*) r_elements_per_layer(i)
  END DO

  H=SUM(hk)                      ! Total Laminate Thickness, H
  z(0)=-H/2.D0                   ! z locations
  DO i=1,N_layers
    z(i)=z(i-1)+hk(i)
  END DO
  READ(6,*) r_layer_loc(0)  ! Inside radius
  READ(6,*) x_length
  lr_factor=x_length/r_layer_loc(0)
  DO i=0,N_layers
    r_layer_loc(i)=r_layer_loc(0)+H/2.D0+z(i)
```

```
  END DO

  r_length=H

  READ(6,*) x_elements

  DO i=1,N_layers
    total_elements_per_layer(i)=r_elements_per_layer(i)*x_elements
  END DO

  layer_temp(1)=total_elements_per_layer(1)
  DO i=2,N_layers
    layer_temp(i)=layer_temp(i-1)+total_elements_per_layer(i)
  END DO

  r_elements=SUM(r_elements_per_layer)

  total_elements=x_elements*r_elements
  NPE=4
  NDF=3
  NGP=2

  x_nodes=x_elements+1
  r_nodes=r_elements+1
  total_nodes=x_nodes*r_nodes

  ALLOCATE(delta_x(total_nodes))
  ALLOCATE(delta_r(total_nodes))
!  ALLOCATE(r_eta(NPE))

  DO i=1,total_nodes
    delta_x(i)=x_length/x_elements
    delta_r(i)=r_length/r_elements
  END DO

  NEQ=total_nodes*NDF

  READ(6,*) n_timesteps
  !WRITE(*,*) " - END SUBROUTINE DIMENSIONS"

END SUBROUTINE GEOMETRY
!--------------------------------------------------------------------------------------------------------------------------------!


!--------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE C_BAR

  USE VARIABLES
```

```
IMPLICIT NONE

!WRITE(*,*) " + BEGIN SUBROUTINE C_BAR ( 2/10)"

Cbar=0.D0

DO k=1,N_layers

  CALL MATERIAL_PROPERTIES_I

  v13=v12

  ! Conjugate Poisson's Ratios
  v21=v12*E2/E1
  v32=v23*E3/E2
  v31=v13*E3/E1

  v=v12*v21+v23*v32+v13*v31+2.D0*v21*v32*v13

  C11=(1.D0-v23*v32)*E1/(1.D0-v)
  C12=(v12+v32*v13)*E2/(1.D0-v)
  C13=(v13+v12*v23)*E3/(1.D0-v)
  C22=(1.D0-v13*v31)*E2/(1.D0-v)
  C23=(v23+v21*v13)*E3/(1.D0-v)
  C33=(1.D0-v12*v21)*E3/(1.D0-v)
  C44=G23
  C55=G13
  C66=G12

  COS_theta=COS(thetak(k)*deg_to_rad)     ! Laminate angle, COS(thetak)
  SIN_theta=SIN(thetak(k)*deg_to_rad)     ! Laminate angle, SIN(thetak)

  Cbar(1,1,k)=COS_theta**4*C11+2.D0*COS_theta**2*SIN_theta**2*(C12+2.D0*C66)+SIN_theta**4*C22
  Cbar(1,2,k)=COS_theta**2*SIN_theta**2*(C11+C22-4.D0*C66)+(SIN_theta**4+COS_theta**4)*C12
  Cbar(1,3,k)=COS_theta**2*C13+SIN_theta**2*C23
  Cbar(1,6,k)=SIN_theta*COS_theta*(COS_theta**2*(C11-C12-2.D0*C66)+SIN_theta**2*(C12-C22+2.D0*C66))
  Cbar(2,2,k)=SIN_theta**4*C11+2.D0*COS_theta**2*SIN_theta**2*(C12+2.D0*C66)+COS_theta**4*C22
  Cbar(2,3,k)=SIN_theta**2*C13+COS_theta**2*C23
  Cbar(2,6,k)=SIN_theta*COS_theta*(SIN_theta**2*(C11-C12-2.D0*C66)+COS_theta**2*(C12-C22+2.D0*C66))
  Cbar(3,3,k)=C33
  Cbar(3,6,k)=COS_theta*SIN_theta*(C13-C23)
  Cbar(4,4,k)=COS_theta**2*C44+SIN_theta**2*C55
  Cbar(4,5,k)=COS_theta*SIN_theta*(C55-C44)
  Cbar(5,5,k)=SIN_theta**2*C44+COS_theta**2*C55
  Cbar(6,6,k)=SIN_theta**2*COS_theta**2*(C11-2.D0*C12+C22)+(SIN_theta**2-COS_theta**2)**2*C66
  Cbar(2,1,k)=Cbar(1,2,k)
  Cbar(3,1,k)=Cbar(1,3,k)
  Cbar(3,2,k)=Cbar(2,3,k)
```

```
    Cbar(6,1,k)=Cbar(1,6,k)
    Cbar(6,2,k)=Cbar(2,6,k)
    Cbar(6,3,k)=Cbar(3,6,k)

    rho(k)=density

  END DO

  !WRITE(*,*) " - END SUBROUTINE C_BAR"

END SUBROUTINE C_BAR
!----------------------------------------------------------------------------------------------------------------------------------!


!----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE NUMBERING

  USE VARIABLES
  IMPLICIT NONE

  !WRITE(*,*) " + BEGIN SUBROUTINE NUMBERING ( 3/10)"

  ALLOCATE(x_position(total_nodes,0:n_timesteps))
  ALLOCATE(r_position(total_nodes,0:n_timesteps))
  ALLOCATE(t_position(total_nodes,0:n_timesteps))
  ALLOCATE(average_x_position(total_nodes))
  ALLOCATE(average_r_position(total_nodes))

  ALLOCATE(node(total_elements,NPE)) ! 4 node element
  node(1,1)=1
  node(1,2)=2
  node(1,3)=x_elements+3
  node(1,4)=node(1,3)-1

  m=1 ! r_elements /= 1
  DO n=2,r_elements
    l=(n-1)*x_elements+1
    DO i=1,NPE
      node(l,i)=node(m,i)+x_nodes
    END DO
    m=l
  END DO

  DO ni=2,x_elements ! x_elements /= 1
    DO i=1,NPE
      node(ni,i)=node(ni-1,i)+1
    END DO
    m=ni
```

```
    DO nj=2,r_elements
      l=(nj-1)*x_elements+ni
      DO j=1,NPE
        node(l,j)=node(m,j)+x_nodes
      END DO
      m=l
    END DO
  END DO

  t_position(:,tt)=0.D0
  yc=r_layer_loc(0) ! r_layer_loc(0)=inside radius
  DO ni=1,r_nodes
    i=(x_nodes)*(ni-1)+1
    j=ni
    x_position(i,tt)=0.D0
    r_position(i,tt)=yc
    DO nj=1,x_elements
      i=i+1
      x_position(i,tt)=x_position(i-1,tt)+delta_x(nj)
      r_position(i,tt)=yc
    END DO
    yc=yc+delta_r(j)
  END DO

  hbw=0
  DO n=1,total_elements
    DO i=1,NPE
      DO j=1,NPE
        nw=(IABS(node(n,i)-node(n,j))+1)*NDF
        IF(hbw<nw) THEN
          hbw=nw
        END IF
      END DO
    END DO
  END DO

  ALLOCATE(t(0:n_timesteps))
  READ(6,*) time
  !READ(6,*) post_muzzle_factor
  !time_post_muzzle=time*post_muzzle_factor
  delta_time=time/REAL(n_timesteps,DBL)
  t(0)=0.D0
  DO i=1,n_timesteps
    t(i)=t(i-1)+delta_time
  END DO

!$$$$$$   alpha=0.5D0     ! CONSTANT AVERAGE ACCELERATION METHOD
  gamma=0.5D0 ! CONSTANT AVERAGE ACCELERATION METHOD (Craig, pg 514)
```

```
  beta=0.25D0 ! CONSTANT AVERAGE ACCELERATION METHOD (Craig, pg 514)

!$$$$$$   a1=alpha*delta_time
!$$$$$$   a2=(1.D0-alpha)*delta_time
!$$$$$$   a3=2.D0/(gamma*delta_time**2)
!$$$$$$   a4=2.D0/(gamma*delta_time)
!$$$$$$   a5=1.D0/gamma-1.D0

  ALLOCATE(global_stiff(NEQ,hbw))
  ALLOCATE(global_force(NEQ,0:n_timesteps))
  ALLOCATE(element_global_coord(total_elements,3))
  ALLOCATE(global_displacement(NEQ,0:n_timesteps))
  ALLOCATE(global_displacement_static(NEQ))
  ALLOCATE(global_velocity(NEQ,0:n_timesteps))
  ALLOCATE(global_acceleration(NEQ,0:n_timesteps))
  ALLOCATE(temp_array(NEQ,0:n_timesteps))
  ALLOCATE(temp_f(NEQ))
  ALLOCATE(temp_d(NEQ))
  ALLOCATE(temp_v(NEQ))
  ALLOCATE(temp_a(NEQ))
  ALLOCATE(temp_m(NEQ,hbw))
  ALLOCATE(temp_matmul(NEQ))
  ALLOCATE(temp_vector(NEQ))

  ALLOCATE(rep_k_matrix(NEQ,hbw))
  ALLOCATE(rep_m_matrix(NEQ,hbw))
  ALLOCATE(rep_kbar_matrix(NEQ,hbw))
  ALLOCATE(rep_unbanded_matrix(NEQ,NEQ))

  ALLOCATE(global_mass(NEQ,hbw))
  ALLOCATE(temp_global_mass1(NEQ,hbw))
  ALLOCATE(temp_global_mass2(NEQ,hbw))
  ALLOCATE(temp_global_mass3(NEQ,hbw))
  ALLOCATE(global_matrix(NEQ,hbw))
  ALLOCATE(temp_diag(NEQ))
  ALLOCATE(temp_matmul1(NEQ))
  ALLOCATE(temp_matmul2(NEQ))
  ALLOCATE(temp_matmul3(NEQ))
  ALLOCATE(temp_vector1(NEQ))
  ALLOCATE(temp_vector3(NEQ))
  ALLOCATE(temp_vector2(NEQ,0:n_timesteps))

  ALLOCATE(temp_matrix1(NEQ,hbw))
  ALLOCATE(temp_matrix2(NEQ,hbw))
  ALLOCATE(temp_matrix3(NEQ,hbw))
  ALLOCATE(temp_matrix4(NEQ,NEQ))
  ALLOCATE(temp_matrix5(NEQ,NEQ+hbw))
  ALLOCATE(temp_matrix6(NEQ,NEQ))
```

```
ALLOCATE(temp_matrix9(NEQ,NEQ))
ALLOCATE(matrix_inverse(NEQ,NEQ))
ALLOCATE(mass_inverse(NEQ,NEQ))
ALLOCATE(stiffness_inverse(NEQ,NEQ))
ALLOCATE(square_stiff(NEQ,NEQ))
ALLOCATE(square_mass(NEQ,NEQ))
ALLOCATE(square_effective_stiff(NEQ,NEQ))
ALLOCATE(square_effective_stiffness(NEQ,NEQ))
ALLOCATE(square_effective_mass(NEQ,NEQ))
ALLOCATE(square_effective_matrix(NEQ,NEQ))
ALLOCATE(check_inverse(NEQ,NEQ))
ALLOCATE(inverse_banded(NEQ,hbw))
ALLOCATE(aa(NEQ,NEQ))
ALLOCATE(cc(NEQ,NEQ))
ALLOCATE(LL(NEQ,NEQ))
ALLOCATE(UU(NEQ,NEQ))
ALLOCATE(bb(NEQ))
ALLOCATE(dd(NEQ))
ALLOCATE(xx(NEQ))
ALLOCATE(square_damp(NEQ,NEQ))

!ALLOCATE(temp_vector(NEQ))
ALLOCATE(temp_Ka_n(NEQ))
ALLOCATE(temp_Kv_n(NEQ))
ALLOCATE(temp_Ku_n(NEQ))

ALLOCATE(effective_global_stiffness(NEQ,hbw))
ALLOCATE(effective_stiffness(NEQ,hbw))
ALLOCATE(effective_global_force(NEQ,0:n_timesteps))

global_stiff=0.D0 ! GSTIF
global_force=0.D0 ! GR
global_mass=0.D0
global_displacement=0.D0
global_displacement_static=0.D0
global_velocity=0.D0
global_acceleration=0.D0
effective_global_stiffness=0.D0
effective_stiffness=0.D0

ALLOCATE(ibdy(NEQ))
ALLOCATE(nodbdy(NEQ))
ALLOCATE(dofbdy(NEQ))
ALLOCATE(vbdy(NEQ))
ALLOCATE(ibf(NEQ))
ALLOCATE(nodbf(NEQ))
ALLOCATE(dofbf(NEQ))
ALLOCATE(vbf(NEQ))
```

```
  READ(6,*) P_constant ! =10000.D0   ! 10 kPa = 1.45 psi

  READ(6,*) V_max



  ALLOCATE(sigma_g_p(NGP,NGP,total_elements,6))
  ALLOCATE(epsilon_g_p(NGP,NGP,total_elements,6))

  ALLOCATE(r_g_p(NGP,NGP,total_elements))
  ALLOCATE(x_g_p(NGP,NGP,total_elements))
  ALLOCATE(x_displacement(total_nodes,0:n_timesteps))
  ALLOCATE(r_displacement(total_nodes,0:n_timesteps))
  ALLOCATE(t_displacement(total_nodes,0:n_timesteps))
  ALLOCATE(x_displacement_static(total_nodes))
  ALLOCATE(r_displacement_static(total_nodes))
  ALLOCATE(t_displacement_static(total_nodes))

!$$$$$$   ALLOCATE(assumed_mode_shape_x(total_nodes))
!$$$$$$   ALLOCATE(assumed_mode_shape_r(total_nodes))
!$$$$$$   ALLOCATE(assumed_mode_shape_t(total_nodes))
!$$$$$$   ALLOCATE(assumed_mode_shape_all(NDF*total_nodes))

  ALLOCATE(nodal_strain(NPE,total_elements,6)) ! nodal_strain(NPE+1,total_elements,6)
  ALLOCATE(nodal_stress(NPE,total_elements,6)) ! nodal_stress(NPE+1,total_elements,6)
  ALLOCATE(average_nodal_stress(total_elements,6))
  ALLOCATE(average_nodal_strain(total_elements,6))

  ALLOCATE(layer_ID(total_elements))

!$$$$$$   nodal_strain=0.D0
!$$$$$$   nodal_stress=0.D0

  x_displacement=0.D0
  t_displacement=0.D0
  r_displacement=0.D0
  x_displacement_static=0.D0
  t_displacement_static=0.D0
  r_displacement_static=0.D0

  ALLOCATE(normal_time_v_vs_t(609),normal_time_p_vs_t(632),normal_position_v_vs_x(657),normal_position_p_vs_x(658))
  ALLOCATE(normal_v_vs_t(609),normal_p_vs_t(632),normal_v_vs_x(657),normal_p_vs_x(658))
  ALLOCATE(P_variable(x_nodes,0:n_timesteps))
  ALLOCATE(p_interpolated(0:n_timesteps),v_interpolated(0:n_timesteps),x_interpolated(0:n_timesteps))
!$$$$$$   ALLOCATE(p_interpolated_x(0:n_timesteps))
  CALL PROFILE
```

```
  P_max=P_constant
  P_variable=0.D0

  p_interpolated=0.D0
  p_interpolated(0)=normal_p_vs_t(1)*P_max
  p_interpolated(n_timesteps)=normal_p_vs_t(632)*P_max
!$$$$$$   p_interpolated_x(0)=normal_p_vs_x(1)*P_max
!$$$$$$   p_interpolated_x(n_timesteps)=normal_p_vs_x(632)*P_max
!$$$$$$   v_interpolated=0.D0
!$$$$$$   v_interpolated(0)=normal_v_vs_t(1)*V_max
!$$$$$$   v_interpolated(n_timesteps)=normal_v_vs_t(609)*V_max
  x_interpolated=0.D0
  x_interpolated(0)=0.D0
  x_interpolated(n_timesteps)=x_length

!$$$$$$   READ(6,*) bullet_mass

!WRITE(*,*) " - END SUBROUTINE NUMBERING"


END SUBROUTINE NUMBERING
!----------------------------------------------------------------------------------------------------------------------------------!


!----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE GLOBAL_MATRICES

  USE VARIABLES
  IMPLICIT NONE

  !WRITE(*,*) " + BEGIN SUBROUTINE GLOBAL_MATRICES ( 4/10)"

  ! Degree of freedom index: x=1, r=2, theta=3

  DO nn=1,total_elements

    IF(nn<=layer_temp(1))THEN
      !layer=1
      layer_ID(nn)=1
    END IF
    DO ii=2,N_layers
      IF(nn>layer_temp(ii-1).AND.nn<=layer_temp(ii))THEN
        !layer=ii
        layer_ID(nn)=ii
      END IF
    END DO

    !tt=0
```

```
    DO ii=1,NPE
      element_global_coord(ii,1)=x_position(node(nn,ii),tt)
      element_global_coord(ii,2)=r_position(node(nn,ii),tt)
      element_global_coord(ii,3)=t_position(node(nn,ii),tt)
    END DO
    CALL ELEMENT_MATRICES
    DO ii=1,NPE
      rowbase=(node(nn,ii)-1)*NDF
      DO idof=1,NDF
        lrow=(ii-1)*NDF+idof
        row=rowbase+idof
        global_force(row,tt)=global_force(row,tt)!+R(lrow)
        DO j=1,NPE
          colbase=(node(nn,j)-1)*NDF
          DO JDOF=1,NDF
            lcol=(j-1)*NDF+jdof
            col=colbase+jdof-row+1
            IF(col.GT.0) THEN
              global_stiff(row,col)=global_stiff(row,col)+ELK(lrow,lcol) ! global stiffness
              global_mass(row,col)=global_mass(row,col)+ELM(lrow,lcol) ! global mass
              !effective_stiffness(row,col)=effective_stiffness(row,col)+ELEFFECTIVE(lrow,lcol)
            END IF
          END DO
        END DO
      END DO
    END DO
  END DO


  !WRITE(*,*) " - END SUBROUTINE GLOBAL_MATRICES"

END SUBROUTINE GLOBAL_MATRICES
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE STEP_0

  USE VARIABLES
  IMPLICIT NONE

  WRITE(*,*) "step 0"
  ! INPUT MASS, DAMPING, AND STIFFNESS MATRICES [M], [C], AND [K]

  CALL GLOBAL_MATRICES ! [M] & [K] banded

  temp_matrix3=global_stiff ! [K] banded
```

```
  CALL UNBAND ! temp_matrix4 is output
  square_stiff=temp_matrix4 ! [K] UN-banded

  temp_matrix3=global_mass ! [M] banded
  CALL UNBAND ! temp_matrix4 is output
  square_mass=temp_matrix4 ! [M] UN-banded

!$$$$$$   temp_matrix3=effective_stiffness ! [K]+a3*[M] banded
!$$$$$$   CALL UNBAND ! temp_matrix4 is output
!$$$$$$   square_effective_stiffness=temp_matrix4 ! [K]+a3*[M] UN-banded



  temp_vector1=global_force(:,tt)         ! input vector updated with boundary conditions
  temp_vector3=global_displacement(:,tt)  ! input vector updated with boundary conditions
  CALL INTERNAL_PRESSURE_BCS
  ! ###############################################################################################
  IF(CASE_FLAG==2) THEN
    CALL VARIABLE_PRESSURE_BCS   ! DYNAMIC TRANSIENT SOLUTION (1/4)
  END IF
  ! ###############################################################################################
  CALL ESSENTIAL_BOUNDARY_CONDITIONS              ! FOR BANDED ONLY (DISPLACEMENTS)
  CALL NATURAL_BOUNDARY_CONDITIONS                ! FOR BANDED ONLY (FORCES)
  global_force(:,tt)=temp_vector1
  global_displacement(:,tt)=temp_vector3


  WRITE(*,*) "Begin Solving Static Problem"

  temp_matrix1=global_stiff
  temp_vector2=global_force ! includes boundary conditions
  CALL SOLVE
  global_displacement_static=temp_vector2(:,tt)

  temp_vector3=global_displacement_static
  CALL ESSENTIAL_BOUNDARY_CONDITIONS        ! FOR BANDED ONLY (DISPLACEMENTS)
  global_displacement_static=temp_vector3

  DO n=1,total_nodes ! group displacements according to DOF
    x_displacement_static(n)=global_displacement_static((n-1)*NDF+1)
    r_displacement_static(n)=global_displacement_static((n-1)*NDF+2)
    t_displacement_static(n)=global_displacement_static((n-1)*NDF+3)
  END DO

  !assumed_mode_shape_x=x_displacement_static/MAXVAL(x_displacement_static)
  !assumed_mode_shape_r=r_displacement_static/MAXVAL(r_displacement_static)
  !assumed_mode_shape_t=t_displacement_static/MAXVAL(t_displacement_static)
```

```
  !assumed_mode_shape_all=global_displacement_static/MAXVAL(global_displacement_static)

  CALL STRESS_STRAIN

!$$$$$$   DO n=1,N_layers
!$$$$$$     IF(materialprop(n)==6) THEN
!$$$$$$       CALL MATERIAL_PROPERTIES_I
!$$$$$$     END IF
!$$$$$$   END DO


  ! ################################################################################################
  IF(CASE_FLAG==1) THEN
    loss_factor=strain_energy_visco/strain_energy*visco_loss_factor2 ! STATIC TRANSIENT SOLUTION (1/3)
    WRITE(40,*) loss_factor
    CLOSE(40)   ! Close file for output
  END IF
  ! ################################################################################################

  ! ################################################################################################
  ! DYNAMIC TRANSIENT SOLUTION (2/4)
  ! THIS VALUE IS CALCULATED FROM THE STATIC TRANSIENT SOLUTION
  IF(CASE_FLAG==2) THEN
   READ(40,*) loss_factor
  END IF
  ! ################################################################################################



  ! ################################################################################################
  IF(CASE_FLAG==1) THEN
    square_damp=0.D0 ! STATIC TRANSIENT SOLUTION (2/3)
  END IF
  ! ################################################################################################

  ! ################################################################################################
  ! DYNAMIC TRANSIENT SOLUTION (3/4)
  IF(CASE_FLAG==2) THEN
    ! VALUES ARE CALCULATED FROM ESTIMATED PERIOD IN DYNAMIC_OUTPUT.CSV
    ! omega1=1.1D0*4509463.1391241D0  ! case 0 Stainless Steel    .223 barrel
      omega1=0.9*1943253.1877875D0  ! case 2 damped   composite .223 barrel
    ! omega2=0.9D0*4509463.1391241D0  ! case 0 Stainless Steel    .223 barrel
    ! omega2=0.9D0*2617993.87799149D0 ! case 1 undamped composite .223 barrel
    ! omega2=0.9D0*1943253.1877875D0  ! case 2 damped   composite .223 barrel
    ! omega2=0.9D0*2631700.65222182D0 ! case 3 damped   composite .223 barrel
    ! omega2=0.9D0*1943253.1877875D0  ! case 4 damped   composite .223 barrel
    ! omega2=0.9D0*1943253.1877875D0  ! case 5 damped   composite .223 barrel
     omega2=1.1D0*2048864.7740803D0  ! case 6 damped   composite .223 barrel
```

```
  ! omega1=1.1D0*2463994.23810964D0 ! case 0 Stainless Steel    .50 BMG barrel
  ! omega2=0.9D0*2463994.23810964D0 ! case 0 Stainless Steel    .50 BMG barrel
  ! omega2=0.9D0*750455.097901414D0 ! case 1 undamped composite .50 BMG barrel
  ! omega2=0.9D0*1206854.32070676D0 ! case 6 damped   composite .50 BMG barrel

  !omega1=132977.466818616D0 ! ss/comp
  !omega2=168902.830838161D0 ! ss/comp/ve



  ! Raleigh w1-w2 ! loss_factor*(square_stiff+square_mass)
  square_damp=((loss_factor*omega1*omega2)/(omega1+omega2))*square_mass+((loss_factor)/(omega1+omega2))*square_stiff

END IF
! ################################################################################################

WRITE(*,*) "Finished Solving Static Problem"

WRITE(9,*)
WRITE(9,*) "-------------------------------------------------------------"
WRITE(9,*) "loss factor"
WRITE(9,*) loss_factor
WRITE(9,*) "strain energy"
WRITE(9,*) strain_energy
WRITE(9,*) "viscoelastic strain energy"
WRITE(9,*) strain_energy_visco


WRITE(*,'(A21,I6,A2,I6)') "Inverting Mass Matrix",NEQ,"x",NEQ
CALL CPU_TIME(T3)

IF(CASE_FLAG==1)THEN
  aa=square_mass ! [M] UN-banded
  CALL INVERSE
  mass_inverse=cc  ! [M]^-1 UN-banded
  DO i=1,NEQ
    DO j=1,NEQ
      WRITE(50,*) mass_inverse(i,j)
    END DO
  END DO
ELSE IF(CASE_FLAG==2) THEN
  DO i=1,NEQ
    DO j=1,NEQ
      READ(50,*) mass_inverse(i,j)
    END DO
  END DO
END IF
CALL CPU_TIME(T4)
```

```
WRITE(*,*) "Mass Matrix inverted"
WRITE(*,*) T4-T3,"seconds to invert mass matrix"

! STEP 0.5  (Craig, pg. 514)
global_acceleration(:,tt)=MATMUL(mass_inverse,(global_force(:,tt)-MATMUL(square_damp,global_velocity(:,tt)) &
                       & -MATMUL(square_stiff,global_displacement(:,tt)))) ! [M]^-1 UN-banded

! STEP 0.6 (Craig, pg. 514)
square_effective_matrix=square_mass+gamma*square_damp+beta*delta_time**2*square_stiff ! [M] & [C] & [K] UN-banded


! need to band damping matrix for this
!effective_global_stiffness=square_mass+gamma*square_damp+beta*delta_time**2*square_stiff ! [M] & [C] & [K] banded

! STEP 0.7 (Craig, pg. 514)
global_displacement(:,tt+1)=global_displacement(:,tt)-delta_time*global_velocity(:,tt)+ &
                          & beta*delta_time**2/2.D0*global_acceleration(:,tt)



WRITE(*,'(A26,I6,A2,I6)') "Inverting Effective Matrix",NEQ,"x",NEQ
CALL CPU_TIME(T5)
IF(CASE_FLAG==1)THEN
  aa=square_effective_matrix ! [K_eff]=[M]+gamma*[C]+beta*h^2*[K] UN-banded
  CALL INVERSE
  matrix_inverse=cc          ! [K_eff]^-1 UN-banded
  DO i=1,NEQ
    DO j=1,NEQ
      WRITE(60,*) matrix_inverse(i,j)
    END DO
  END DO
  ELSE IF(CASE_FLAG==2) THEN
  DO i=1,NEQ
    DO j=1,NEQ
      READ(60,*) matrix_inverse(i,j)
    END DO
  END DO
END IF
CALL CPU_TIME(T6)
WRITE(*,*) "Effective Matrix inverted"
WRITE(*,*) T6-T5,"seconds to invert effective matrix"

effective_global_force(:,tt)=global_force(:,tt)

! STEP 1 (Craig, pg. 514)
WRITE(*,*) "Begin Newmark Beta Method"
DO tt=1,n_timesteps
```

```
  ! ##################################################################################################
  IF(CASE_FLAG==2) THEN
    CALL VARIABLE_PRESSURE_BCS  !  DYNAMIC TRANSIENT SOLUTION (4/4)
  END IF
  ! ##################################################################################################

  temp_vector1=global_force(:,tt)         ! input vector updated with boundary conditions
  !temp_matrix1=effective_global_stiffness ! (BANDED)
  CALL NATURAL_BOUNDARY_CONDITIONS        ! FOR BANDED ONLY (FORCES)
  global_force(:,tt)=temp_vector1


  ! MATMUL([A],{x})={B}_column vector
  ! MATMUL({x},[A])={B}_row vector

  ! STEP 2 (Craig, pg. 514)
  effective_global_force(:,tt)=global_force(:,tt)-MATMUL(square_stiff,global_displacement(:,tt-1)) &
                                             & -MATMUL((square_damp+delta_time*square_stiff),global_velocity(:,tt-1)) &
   & -MATMUL((delta_time*(1.D0-gamma)*square_damp+delta_time**2*(1.D0-2.D0*beta)/2.D0*square_stiff),global_acceleration(:,tt-1))

  ! STEP 3 (Craig, pg. 514)
  global_acceleration(:,tt)=MATMUL(matrix_inverse,effective_global_force(:,tt))

  ! STEP 4 (Craig, pg. 514)
  global_displacement(:,tt)=global_displacement(:,tt-1)+delta_time*global_velocity(:,tt-1)+ &
     & ((1.D0-2.D0*beta)*global_acceleration(:,tt-1)+2.D0*beta*global_acceleration(:,tt))*delta_time**2/2.D0

  global_velocity(:,tt)=global_velocity(:,tt-1)+((1.D0-gamma)*global_acceleration(:,tt-1)+ &
     & gamma*global_acceleration(:,tt))*delta_time



  temp_vector3=global_displacement(:,tt)    ! input vector updated with boundary conditions
  CALL ESSENTIAL_BOUNDARY_CONDITIONS        ! FOR BANDED ONLY (DISPLACEMENTS)
  global_displacement(:,tt)=temp_vector3


DO n=1,total_nodes ! group displacements according to DOF
  x_displacement(n,tt)=global_displacement((n-1)*NDF+1,tt)
  t_displacement(n,tt)=global_displacement((n-1)*NDF+3,tt)
  r_displacement(n,tt)=global_displacement((n-1)*NDF+2,tt)
END DO
DO n=1,total_nodes
  x_position(n,tt)=x_position(n,tt-1)+x_displacement(n,tt)
  t_position(n,tt)=t_position(n,tt-1)+t_displacement(n,tt)
  r_position(n,tt)=r_position(n,tt-1)+r_displacement(n,tt)
```

```
  END DO


    WRITE(*,'(A5,I7,F6.1,A7)')"time",tt,REAL(tt,DBL)/REAL(n_timesteps,DBL)*100.0D0," % done"
    IF((tt>0).AND.((global_displacement((x_elements/2+1)*NDF+3,tt)/r_layer_loc(0)>10.D0)))THEN
      ttt=tt
      GOTO 1000
    ELSE IF (tt==n_timesteps)THEN
      ttt=n_timesteps
    END IF



  END DO
  WRITE(*,*) "End Newmark Beta Method"

  1000 WRITE(*,*)
  IF(ttt<n_timesteps-1) WRITE(*,*) "IT BLEW UP"

END SUBROUTINE STEP_0
!------------------------------------------------------------------------------------------------------------------------------------!

!------------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE STRESS_STRAIN

  USE VARIABLES
  IMPLICIT NONE

  !WRITE(*,*) " + BEGIN SUBROUTINE STRESS_STRAIN ( 7/10)"

  nodal_strain=0.D0
  nodal_stress=0.D0
  average_nodal_stress=0.D0
  average_nodal_strain=0.D0
  sigma_g_p=0.D0
  epsilon_g_p=0.D0
  strain_energy=0.D0
  strain_energy_visco=0.D0
!$$$$$$   counter=1
!$$$$$$   layer=1

  DO n=1,total_elements
    DO iii=1,NPE
      element_global_coord(iii,1)=x_position(node(n,iii),tt)
      element_global_coord(iii,2)=r_position(node(n,iii),tt)
      element_global_coord(iii,3)=t_position(node(n,iii),tt)
    END DO
    !Differentiate displacements with shape function derivatives and definitions to calculate gauss point strains
```

```
DO igp=1,NGP
  xi=GAUSS(igp,NGP)
  DO jgp=1,NGP
    eta=GAUSS(jgp,NGP)
    CALL SHAPE_FUNCTIONS
    r_eta=0.D0
    x_xi=0.D0
    r_g_p=0.D0
    x_g_p=0.D0
    u_x_g_p=0.D0
    u_theta_g_p=0.D0
    u_r_g_p=0.D0
    du_x_dx=0.D0
    du_x_dr=0.D0
    du_theta_dx=0.D0
    du_theta_dr=0.D0
    du_r_dx=0.D0
    du_r_dr=0.D0
    !Interpolate derivatives, positions, and displacements at the current gauss point
    DO i=1,npe
      x_xi        = x_xi        + element_global_coord(i,1)*NSF(i)      !element_global_coord => x=1, r=2, t=3
      r_eta       = r_eta       + element_global_coord(i,2)*NSF(i)
      u_theta_g_p = u_theta_g_p + NSF(i)*t_displacement_static(node(n,i))
      u_x_g_p     = u_x_g_p     + NSF(i)*x_displacement_static(node(n,i))
      u_r_g_p     = u_r_g_p     + NSF(i)*r_displacement_static(node(n,i))
      du_x_dx     = du_x_dx     + gdSF(1,i)*x_displacement_static(node(n,i))
      du_x_dr     = du_x_dr     + gdSF(2,i)*x_displacement_static(node(n,i))
      du_theta_dx = du_theta_dx + gdSF(1,i)*t_displacement_static(node(n,i))
      du_theta_dr = du_theta_dr + gdSF(2,i)*t_displacement_static(node(n,i))
      du_r_dx     = du_r_dx     + gdSF(1,i)*r_displacement_static(node(n,i))
      du_r_dr     = du_r_dr     + gdSF(2,i)*r_displacement_static(node(n,i))
    END DO
    !Gauss point positions
    r_g_p(igp,jgp,n)=r_eta
    x_g_p(igp,jgp,n)=x_xi
    !Calculate strains from kinematic definitions and subtract off free thermal strains (u=x, v=theta, w=r)
    epsilon_g_p(igp,jgp,n,1)=du_x_dx                        ! epsilon_x
    epsilon_g_p(igp,jgp,n,2)=u_r_g_p/r_eta                  ! epsilon_theta
    epsilon_g_p(igp,jgp,n,3)=du_r_dr                        ! epsilon_r
    epsilon_g_p(igp,jgp,n,4)=du_theta_dr-u_theta_g_p/r_eta  ! gamma_theta_r
    epsilon_g_p(igp,jgp,n,5)=du_x_dr+du_r_dx                ! gamma_r_x
    epsilon_g_p(igp,jgp,n,6)=du_theta_dx                    ! gamma_x_theta
    !Calculate the gauss point stresses from the constitutive relationship
    sigma_g_p(igp,jgp,n,1)=Cbar(1,1,layer_ID(n))*epsilon_g_p(igp,jgp,n,1)+&
      Cbar(1,2,layer_ID(n))*epsilon_g_p(igp,jgp,n,2)+Cbar(1,3,layer_ID(n))*epsilon_g_p(igp,jgp,n,3)+&
      Cbar(1,6,layer_ID(n))*epsilon_g_p(igp,jgp,n,6)
    sigma_g_p(igp,jgp,n,2)=Cbar(2,1,layer_ID(n))*epsilon_g_p(igp,jgp,n,1)+&
      Cbar(2,2,layer_ID(n))*epsilon_g_p(igp,jgp,n,2)+Cbar(2,3,layer_ID(n))*epsilon_g_p(igp,jgp,n,3)+&
```

```
          Cbar(2,6,layer_ID(n))*epsilon_g_p(igp,jgp,n,6)
        sigma_g_p(igp,jgp,n,3)=Cbar(3,1,layer_ID(n))*epsilon_g_p(igp,jgp,n,1)+&
          Cbar(3,2,layer_ID(n))*epsilon_g_p(igp,jgp,n,2)+Cbar(3,3,layer_ID(n))*epsilon_g_p(igp,jgp,n,3)+&
          Cbar(3,6,layer_ID(n))*epsilon_g_p(igp,jgp,n,6)
        sigma_g_p(igp,jgp,n,4)=Cbar(4,4,layer_ID(n))*epsilon_g_p(igp,jgp,n,4)+&
          Cbar(4,5,layer_ID(n))*epsilon_g_p(igp,jgp,n,5)
        sigma_g_p(igp,jgp,n,5)=Cbar(4,5,layer_ID(n))*epsilon_g_p(igp,jgp,n,4)+&
          Cbar(5,5,layer_ID(n))*epsilon_g_p(igp,jgp,n,5)
        sigma_g_p(igp,jgp,n,6)=Cbar(6,1,layer_ID(n))*epsilon_g_p(igp,jgp,n,1)+&
          Cbar(6,2,layer_ID(n))*epsilon_g_p(igp,jgp,n,2)+Cbar(6,3,layer_ID(n))*epsilon_g_p(igp,jgp,n,3)+&
          Cbar(6,6,layer_ID(n))*epsilon_g_p(igp,jgp,n,6)
!        WRITE(9,'(4X,I5,2(9X,I2),2(8X,ES12.4))') n,igp,jgp,x_g_p(igp,jgp,n),r_g_p(igp,jgp,n)



        strain_energy=strain_energy+epsilon_g_p(igp,jgp,n,3)*sigma_g_p(igp,jgp,n,3)/2.D0 ! x=1, t=2, r=3
      IF(materialprop(layer_ID(n))==6) THEN
        strain_energy_visco=strain_energy_visco+epsilon_g_p(igp,jgp,n,3)*sigma_g_p(igp,jgp,n,3)/2.D0
      ELSE
        strain_energy_visco=strain_energy_visco+0.D0
      END IF

      END DO
    END DO
  END DO

  !Find stresses and strains at the corner nodes
  root_3=SQRT(3.D0)
  DO n=1,total_elements
    !Set up local coordinates elxtr = 0.D0
    DO iii=1,NPE
      element_global_coord(iii,1)=x_position(node(n,iii),tt)
      element_global_coord(iii,2)=r_position(node(n,iii),tt)
      element_global_coord(iii,3)=t_position(node(n,iii),tt)
    END DO
    DO jjj=1,NPE  ! NPE+1 FOR Q9
      !Assign xi,eta, depending on which nodal stress is being extrapolated
      IF(jjj==1) THEN
        xi  = -root_3
        eta = -root_3
      ELSE IF(jjj==2) THEN
        xi  =  root_3
        eta = -root_3
      ELSE IF(jjj==3) THEN
        xi  =  root_3
        eta =  root_3
      ELSE IF(jjj==4) THEN
        xi  = -root_3
```

```
        eta =  root_3
      END IF
      CALL SHAPE_FUNCTIONS
      !Extrapolate gauss point stresses to corner nodes (Folkman method)
      !4-node or 8-node element
      nodal_stress(jjj,n,:)=NSF(1)*sigma_g_p(1,1,n,:)+NSF(2)*sigma_g_p(1,2,n,:)+NSF(3)*sigma_g_p(2,1,n,:)+ &
                      & NSF(4)*sigma_g_p(2,2,n,:)
      !Calculate strains from the constitutive relationship
      nodal_strain(jjj,n,:)=NSF(1)*epsilon_g_p(1,1,n,:)+NSF(2)*epsilon_g_p(1,2,n,:)+NSF(3)*epsilon_g_p(2,1,n,:)+ &
                      & NSF(4)*epsilon_g_p(2,2,n,:)

       !WRITE(9,'(I5,I5,6EN13.3)') n,jjj,nodal_strain(jjj,n,:)
    END DO
  END DO

  DO iii=1,total_elements
    average_nodal_stress(iii,:)=(nodal_stress(1,iii,:)+nodal_stress(2,iii,:)+nodal_stress(3,iii,:)+nodal_stress(4,iii,:))/4.D0
    average_nodal_strain(iii,:)=(nodal_strain(1,iii,:)+nodal_strain(2,iii,:)+nodal_strain(3,iii,:)+nodal_strain(4,iii,:))/4.D0
    average_r_position(iii)=(r_position(node(iii,1),tt)+r_position(node(iii,2),tt)+r_position(node(iii,3),tt)+ &
          & r_position(node(iii,4),tt))/4.D0
    average_x_position(iii)=(x_position(node(iii,1),tt)+x_position(node(iii,2),tt)+x_position(node(iii,3),tt)+ &
          & x_position(node(iii,4),tt))/4.D0
  END DO



!$$$$$$   DO i=1,6 ! 1=x, 2=t, 3=r
!$$$$$$     jjj=5!DO jjj=1,x_elements
!$$$$$$       WRITE(9,*)
!$$$$$$       WRITE(9,*) "Element",",","Nodal Stress",",","R Position",",","X Position"
!$$$$$$       DO iii=jjj,total_elements,x_elements
!$$$$$$         WRITE(9,*) iii,",",nodal_stress(1,iii,i),",",r_position(node(iii,1),tt),",",x_position(node(iii,1),tt)  !~(NPE (1-
4),element,1-6)
!$$$$$$         WRITE(9,*) iii,",",nodal_stress(4,iii,i),",",r_position(node(iii,4),tt),",",x_position(node(iii,4),tt)  !~(NPE (1-
4),element,1-6)
!$$$$$$       END DO
!$$$$$$       WRITE(9,*)
!$$$$$$       WRITE(9,*) "Element",",","Nodal Stress",",","R Position",",","X Position"
!$$$$$$       DO iii=jjj,total_elements,x_elements
!$$$$$$         WRITE(9,*) iii,",",nodal_stress(2,iii,i),",",r_position(node(iii,2),tt),",",x_position(node(iii,2),tt)  !~(NPE (1-
4),element,1-6)
!$$$$$$         WRITE(9,*) iii,",",nodal_stress(3,iii,i),",",r_position(node(iii,3),tt),",",x_position(node(iii,3),tt)  !~(NPE (1-
4),element,1-6)
!$$$$$$       END DO
!$$$$$$     !END DO
!$$$$$$   END DO

  DO i=1,6 ! 1=x, 2=t, 3=r
```

```
  DO jjj=1,x_elements
    IF(i==1) WRITE(9,*) "Sigma X"
    IF(i==2) WRITE(9,*) "Sigma Theta"
    IF(i==3) WRITE(9,*) "Sigma R"
    IF(i==4) WRITE(9,*) "Tau R-Theta"
    IF(i==5) WRITE(9,*) "Tau X-R"
    IF(i==6) WRITE(9,*) "Tau X-Theta"
    WRITE(9,*) "Average Nodal Stress",",","R Position",",","X Position"
    DO iii=jjj,total_elements,x_elements
      WRITE(9,*) average_nodal_stress(iii,i),",",average_r_position(iii),",",average_x_position(iii)  !~(NPE (1-4),element,1-6)
    END DO
    WRITE(9,*)
  END DO
END DO

WRITE(9,*)
WRITE(9,*) "-----------------------------------------------------------"
WRITE(9,*)

DO i=1,6 ! 1=x, 2=t, 3=r
  DO jjj=1,x_elements
    IF(i==1) WRITE(9,*) "Epsilon X"
    IF(i==2) WRITE(9,*) "Epsilon Theta"
    IF(i==3) WRITE(9,*) "Epsilon R"
    IF(i==4) WRITE(9,*) "Gamma R-Theta"
    IF(i==5) WRITE(9,*) "Gamma X-R"
    IF(i==6) WRITE(9,*) "Gamma X-Theta"
    WRITE(9,*) "Average Nodal Strain",",","R Position",",","X Position"
    DO iii=jjj,total_elements,x_elements
      WRITE(9,*) average_nodal_strain(iii,i),",",average_r_position(iii),",",average_x_position(iii)  !~(NPE (1-4),element,1-6)
    END DO
    WRITE(9,*)
  END DO
END DO


WRITE(9,*)
WRITE(9,*) "-----------------------------------------------------------"
WRITE(9,*)


DO i=1,6 ! 1=x, 2=t, 3=r
  DO jjj=1,x_elements
    IF(i==1) WRITE(9,*) "Sigma X vs Epsilon X"
    IF(i==2) WRITE(9,*) "Sigma Theta vs Epsilon Theta"
    IF(i==3) WRITE(9,*) "Sigma R vs Epsilon R"
    IF(i==4) WRITE(9,*) "Tau R-Theta vs Gamma R-Theta"
    IF(i==5) WRITE(9,*) "Tau X-R vs Gamma X-R"
```

```
      IF(i==6) WRITE(9,*) "Tau X-Theta vs Gamma X-Theta"
      !WRITE(9,*) "Average Nodal Strain",",","Average Nodal Stress"
      WRITE(9,*) "Nodal Strain",",","Nodal Stress"
      DO iii=jjj,total_elements,x_elements
        !WRITE(9,*) average_nodal_strain(iii,i),",",average_nodal_stress(iii,i)
        WRITE(9,*) nodal_strain(1,iii,i),",",nodal_stress(1,iii,i),",","Node 1"  !~(NPE (1-4),element,1-6)
        WRITE(9,*) nodal_strain(4,iii,i),",",nodal_stress(4,iii,i),",","Node 4"  !~(NPE (1-4),element,1-6)
      END DO
      DO iii=jjj,total_elements,x_elements
        !WRITE(9,*) average_nodal_strain(iii,i),",",average_nodal_stress(iii,i)
        WRITE(9,*) nodal_strain(2,iii,i),",",nodal_stress(2,iii,i),",","Node 2"  !~(NPE (1-4),element,1-6)
        WRITE(9,*) nodal_strain(3,iii,i),",",nodal_stress(3,iii,i),",","Node 3"  !~(NPE (1-4),element,1-6)
      END DO
      WRITE(9,*)
    END DO
  END DO


!$$$$$$     DO jjj=1,x_elements
!$$$$$$       DO iii=jjj,total_elements,x_elements
!$$$$$$         WRITE(9,*) (nodal_stress(1,iii,3)+nodal_stress(2,iii,3)+nodal_stress(3,iii,3)+nodal_stress(4,iii,3))/4.D0,",", &
!$$$$$$                  &
(r_position(node(iii,1),tt)+r_position(node(iii,2),tt)+r_position(node(iii,3),tt)+r_position(node(iii,4),tt))/4.D0,",", &
!$$$$$$                  &
(x_position(node(iii,1),tt)+x_position(node(iii,2),tt)+x_position(node(iii,3),tt)+x_position(node(iii,4),tt))/4.D0  !~(NPE (1-
4),element,1-6)
!$$$$$$       END DO
!$$$$$$     END DO
!$$$$$$     WRITE(9,*)


!$$$$$$ WRITE(9,*) "time = ",tt
!$$$$$$ WRITE(9,*) "node # =",
!$$$$$$ WRITE(9,*) "middle, x (1)"
!$$$$$$ WRITE(9,*) "middle, t (2)"
!$$$$$$ WRITE(9,*) "middle, r (3)"
!$$$$$$ WRITE(9,*) "middle, tr (4)"
!$$$$$$ WRITE(9,*) "middle, xr (5)"
!$$$$$$ WRITE(9,*) "middle, tx (6)"


  WRITE(*,*) " - END SUBROUTINE STRESS_STRAIN"
```

```
END SUBROUTINE STRESS_STRAIN
!------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE ESSENTIAL_BOUNDARY_CONDITIONS ! (DISPLACEMENTS)

  USE VARIABLES
  IMPLICIT NONE



  ! apply displacement essential boundary conditions
  IF(nbdy/=0) THEN
    DO n=1,nbdy
      nb=ibdy(n)
      temp_vector3(nb)=vbdy(n)
    END DO
  END IF



END SUBROUTINE ESSENTIAL_BOUNDARY_CONDITIONS
!------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE NATURAL_BOUNDARY_CONDITIONS ! (FORCES)

  USE VARIABLES
  IMPLICIT NONE

  !WRITE(*,*) " + BEGIN SUBROUTINE BOUNDARY_CONDITION ( 5/10)"

  ! input Boundary conditions



  ! apply force natural boundary conditions
  IF(nbf/=0) THEN
    DO n=1,nbf
      nb=ibf(n)
      temp_vector1(nb)=vbf(n) ! temp_vector1(nb)+vbf(n)
    END DO
  END IF
```

```
END SUBROUTINE NATURAL_BOUNDARY_CONDITIONS
!-----------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE BANDED_MULT

  USE VARIABLES
  IMPLICIT NONE

  temp_matmul1=0
  temp_matmul2=0
  temp_matmul3=0
  ! UPPER DIAGONAL
  counter=1
  DO iiii=1,NEQ
    IF(iiii<=NEQ-hbw+1)THEN
        DO jjjj=1,hbw
          temp_matmul1(iiii)=temp_matmul1(iiii)+temp_matrix1(iiii,jjjj)*temp_vector(jjjj+iiii-1)
        END DO
      ELSE IF(iiii>NEQ-hbw+1)THEN
        DO jjjj=1,hbw-counter
          temp_matmul1(iiii)=temp_matmul1(iiii)+temp_matrix1(iiii,jjjj)*temp_vector(jjjj+iiii-1)
        END DO
        counter=counter+1
      END IF
  END DO
  ! LOWER DIAGONAL
  !temp_global_mass1=temp_matrix1
  DO i=1,neq
    DO j=1,hbw
      temp_matrix2(i,j)=temp_matrix1(i,hbw+1-j)
    END DO
  END DO
  temp_matrix3=0
  counter=1
  DO j=1,hbw
    DO i=1,neq
      IF(i+hbw-counter<=neq) temp_matrix3(i+hbw-counter,j)=temp_matrix2(i,j)
    END DO
    counter=counter+1
  END DO
  temp_matrix1=temp_matrix3
  counter=1
  DO i=1,hbw-1
    DO j=1,counter
      temp_matrix3(i,j)=temp_matrix1(i,j+hbw-i)
      temp_matrix3(i,j+hbw-i)=0
```

```
    END DO
    counter=counter+1
  END DO
  temp_diag(:)=temp_matrix3(:,hbw)
  DO i=1,neq
    temp_matrix3(i,hbw)=temp_matrix3(i,hbw)-temp_diag(i)
  END DO
  DO i=1,hbw
    temp_matrix3(i,i)=0
  END DO
  ! ADD LOWER TO UPPER
  counter=1
  DO iiii=1,NEQ
    IF(iiii>=hbw)THEN
        DO jjjj=1,hbw
          temp_matmul3(iiii)=temp_matmul3(iiii)+temp_matrix3(iiii,jjjj)*temp_vector(jjjj+iiii-hbw)
        END DO
      ELSE IF(iiii<hbw)THEN
        DO jjjj=1,counter
          temp_matmul3(iiii)=temp_matmul3(iiii)+temp_matrix3(iiii,jjjj)*temp_vector(jjjj)
        END DO
        counter=counter+1
      END IF
  END DO

  temp_matmul2=temp_matmul1+temp_matmul3

END SUBROUTINE BANDED_MULT
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE SOLVE

  USE VARIABLES
  IMPLICIT NONE

  !WRITE(*,*) " + BEGIN SUBROUTINE SOLVE ( 6/10)"
  !temp_vector2(:,tt)=0.D0
  meqns=NEQ-1
  DO npiv=1,meqns
    npivot=npiv+1
    lstsub=npiv+hbw-1
    IF(lstsub>neq) THEN
      lstsub=neq
    END IF
    DO nrow=npivot,lstsub
      ncol=nrow-npiv+1
```

```
      factor=temp_matrix1(npiv,ncol)/temp_matrix1(npiv,1)
      DO ncol=nrow,lstsub
        icol=ncol-nrow+1
        jcol=ncol-npiv+1
        temp_matrix1(nrow,icol)=temp_matrix1(nrow,icol)-factor*temp_matrix1(npiv,jcol)
      END DO
      temp_vector2(nrow,tt)=temp_vector2(nrow,tt)-factor*temp_vector2(npiv,tt)
    END DO
    !WRITE(*,*) "    ", REAL(npiv,DBL)/REAL(meqns,DBL)*100.D0, "% Done, (1/2)"
  END DO

  DO ijk=2,NEQ
    npiv=NEQ-ijk+2
    temp_vector2(npiv,tt)=temp_vector2(npiv,tt)/temp_matrix1(npiv,1)
    lstsub=npiv-hbw+1
    IF(lstsub<1) THEN
      lstsub=1
    END IF
    npivot=npiv-1
    DO jki=lstsub,npivot
      nrow=npivot-jki+lstsub
      ncol=npiv-nrow+1
      factor=temp_matrix1(nrow,ncol)
      temp_vector2(nrow,tt)=temp_vector2(nrow,tt)-factor*temp_vector2(npiv,tt)
    END DO
    !WRITE(*,*) "    ", REAL(ijk,DBL)/REAL(NEQ,DBL)*100.D0, "% Done, (2/2)"
  END DO
  temp_vector2(1,tt)=temp_vector2(1,tt)/temp_matrix1(1,1)


END SUBROUTINE SOLVE
!------------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE ELEMENT_MATRICES

  USE VARIABLES
  IMPLICIT NONE

  !R=0.D0
  ELK=0.D0
  ELKT=0.D0
  ELF=0.D0
  TK=0.D0
  ELM=0.D0
  ELMT=0.D0
```

```
  ELEFFECTIVE=0.D0
  TM=0.D0

  CALL GAUSS_POINTS

  DO igp=1,NGP
    xi=gauss(igp,NGP)
    DO jgp=1,NGP
      eta=gauss(jgp,NGP)
      CALL SHAPE_FUNCTIONS
      r_eta=0.D0
      DO ii=1,NPE
        r_eta=r_eta+element_global_coord(ii,2)*NSF(ii)
      END DO
!      P_0=P_constant*weight(igp,NGP)*weight(jgp,NGP)*det       ! *2.D0*pi
      CB11=Cbar(1,1,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB12=Cbar(1,2,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB13=Cbar(1,3,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB16=Cbar(1,6,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB22=Cbar(2,2,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB23=Cbar(2,3,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB26=Cbar(2,6,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB33=Cbar(3,3,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB36=Cbar(3,6,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB44=Cbar(4,4,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB45=Cbar(4,5,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB55=Cbar(5,5,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      CB66=Cbar(6,6,layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      rhok=rho(layer_ID(nn))*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi !rho(layer)*weight(igp,NGP)*weight(jgp,NGP)*det*2.D0*pi
      DO i=1,NPE

        ! ELF(i)=ELF(i)+NSF(i)*P_0    ! Force Vector

          dpsi_i_dx=gdSF(1,i) ! dpsi_i_dx=d(psi_i)/dx
          dpsi_i_dr=gdSF(2,i) ! dpsi_i_dr=d(psi_i)/dr

        DO j=1,NPE

          dpsi_j_dx=gdSF(1,j) ! dpsi_j_dx=d(psi_j)/dx
          dpsi_j_dr=gdSF(2,j) ! dpsi_j_dr=d(psi_j)/dr

          TK(1,1,i,j)=TK(1,1,i,j)+(CB11*dpsi_j_dx*dpsi_i_dx*r_eta+CB55*dpsi_j_dr*dpsi_i_dr*r_eta) ! Degree of freedom index: x=1,
theta=3, r=2
          TK(1,3,i,j)=TK(1,3,i,j)+(CB16*dpsi_j_dx*dpsi_i_dx*r_eta+CB45*r_eta*(dpsi_j_dr*dpsi_i_dr-dpsi_i_dr*(NSF(j)/ &
                                & r_eta)))
          TK(1,2,i,j)=TK(1,2,i,j)+((CB12)*dpsi_i_dx*NSF(j)+CB13*r_eta*dpsi_i_dx*dpsi_j_dr+CB55*dpsi_i_dr*dpsi_j_dx*r_eta)
          TK(3,1,i,j)=TK(3,1,i,j)+(CB16*dpsi_j_dx*dpsi_i_dx*r_eta+CB45*r_eta*(dpsi_j_dr*dpsi_i_dr-(NSF(i)/r_eta)* &
                                & dpsi_j_dr))
```

```
          TK(3,3,i,j)=TK(3,3,i,j)+(CB66*r_eta*dpsi_j_dx*dpsi_i_dx+CB44*r_eta*(dpsi_j_dr*dpsi_i_dr-dpsi_j_dr*(NSF(i)/ &
                              & r_eta)-dpsi_i_dr*(NSF(j)/r_eta)+(NSF(j)*NSF(i))/r_eta**2))
          TK(3,2,i,j)=TK(3,2,i,j)+((CB26)*dpsi_i_dx*NSF(j)+CB36*r_eta*dpsi_i_dx*dpsi_j_dr+CB45*r_eta*(dpsi_i_dr*dpsi_j_dx- &
                              & dpsi_j_dx*(NSF(i)/r_eta)))
          TK(2,1,i,j)=TK(2,1,i,j)+(CB55*dpsi_i_dx*dpsi_j_dr*r_eta+CB13*dpsi_i_dr*dpsi_j_dx*r_eta+(CB12)*dpsi_j_dx*NSF(i))
          TK(2,3,i,j)=TK(2,3,i,j)+(CB45*r_eta*(dpsi_i_dx*dpsi_j_dr-dpsi_i_dx*(NSF(j)/r_eta))+CB36*r_eta*dpsi_i_dr* &
                              & dpsi_j_dx+(CB26)*dpsi_j_dx*NSF(i))
          TK(2,2,i,j)=TK(2,2,i,j)+(CB55*r_eta*dpsi_j_dx*dpsi_i_dx+(CB23)*(dpsi_j_dr*NSF(i)+dpsi_i_dr*NSF(j))+CB33*r_eta* &
                              & dpsi_j_dr*dpsi_i_dr+(CB22/r_eta)*NSF(i)*NSF(j))
          TM(1,1,i,j)=TM(1,1,i,j)+rhok*r_eta*NSF(i)*NSF(j)  ! Degree of freedom index: x=1, theta=3, r=2   !
TM(1,1,i,j)+rhok*r_eta*NSF(i)*NSF(j)
          TM(2,2,i,j)=TM(1,1,i,j) ! +rhok*r_eta*NSF(i)*NSF(j)
          TM(3,3,i,j)=TM(1,1,i,j)
        END DO

      END DO
    END DO
  END DO


  DO m=1,NPE ! stiffness terms are placed in one large matrix ordered according to node
    DO n=1,NPE
      DO i=1,NDF
        DO j=1,NDF
          l=(m-1)*NDF+i
          k=(n-1)*NDF+j
          ELK(l,k)=TK(i,j,m,n)!ELKT(m,n,i,j)
          ELM(l,k)=TM(i,j,m,n)!ELMT(m,n,i,j)
          !ELEFFECTIVE(l,k)=ELK(l,k)+a3*ELM(l,k)
        END DO
      END DO
    END DO
  END DO

END SUBROUTINE ELEMENT_MATRICES
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE GAUSS_POINTS

  USE VARIABLES
  IMPLICIT NONE

  ! Gauss points
  gauss=0.D0
  gauss(1,2)=-0.57735026918962576450914870D0
  gauss(1,3)=-0.77459666924148337703585307D0
```

```
  gauss(2,2)= 0.5773502691896257645091487D0
  gauss(3,3)= 0.7745966692414833770358530D0

  ! Weights
  weight=0.D0
  weight(1,1)=2.D0
  weight(1,2)=1.D0
  weight(2,2)=1.D0
  weight(1,3)=0.5555555555555555555555556D0
  weight(2,3)=0.8888888888888888888888889D0
  weight(3,3)=0.5555555555555555555555556D0


END SUBROUTINE GAUSS_POINTS
!------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE SHAPE_FUNCTIONS

  USE VARIABLES
  IMPLICIT NONE

  ! Shape functions
  NSF=0.D0
  NSF(1)=(1.D0-xi)*(1.D0-eta)/4.D0 ! N1
  NSF(2)=(1.D0+xi)*(1.D0-eta)/4.D0 ! N2
  NSF(3)=(1.D0+xi)*(1.D0+eta)/4.D0 ! N3
  NSF(4)=(1.D0-xi)*(1.D0+eta)/4.D0 ! N4

  ! Derivative of shape functions
  dNSF=0.D0
  dNSF(1,1)=(-1.D0+eta)/4.D0 ! d(N1)/d(xi)
  dNSF(1,2)=( 1.D0-eta)/4.D0 ! d(N2)/d(xi)
  dNSF(1,3)=( 1.D0+eta)/4.D0 ! d(N3)/d(xi)
  dNSF(1,4)=(-1.D0-eta)/4.D0 ! d(N4)/d(xi)
  dNSF(2,1)=(-1.D0+xi )/4.D0 ! d(N1)/d(eta)
  dNSF(2,2)=(-1.D0-xi )/4.D0 ! d(N2)/d(eta)
  dNSF(2,3)=( 1.D0+xi )/4.D0 ! d(N3)/d(eta)
  dNSF(2,4)=( 1.D0-xi )/4.D0 ! d(N4)/d(eta)

  !Jacobian matrix
  Jac=0.D0
  DO i=1,2
    DO j=1,2
      DO k=1,NPE
        Jac(i,j)=Jac(i,j)+dNSF(i,k)*element_global_coord(k,j)
      END DO
```

```
    END DO
  END DO

  det=Jac(1,1)*Jac(2,2)-Jac(1,2)*Jac(2,1)

  Jstar(1,1)= Jac(2,2)/det
  Jstar(1,2)=-Jac(1,2)/det
  Jstar(2,1)=-Jac(2,1)/det
  Jstar(2,2)= Jac(1,1)/det

  gdSF=0.D0
  DO i=1,2
    DO j=1,NPE
      DO k=1,2
        gdSF(i,j)=gdSF(i,j)+Jstar(i,k)*dNSF(k,j)
      END DO
    END DO
  END DO

END SUBROUTINE SHAPE_FUNCTIONS
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE MATERIAL_PROPERTIES_I

  USE VARIABLES
  IMPLICIT NONE

  IF     (materialprop(k)==1) THEN
    CALL GRAPHITE_POLYMER            ! Call material properties - 1) Graphite Polymer
  ELSE IF(materialprop(k)==2) THEN
    CALL T300_5280
  ELSE IF(materialprop(k)==3) THEN
    CALL STAINLESS_STEEL_ENG
  ELSE IF(materialprop(k)==4) THEN
    CALL STAINLESS_STEEL_SI
  ELSE IF(materialprop(k)==5) THEN
    CALL FRONK_GRAPHITE_POLYMER
  ELSE IF(materialprop(k)==6) THEN
    CALL VISCO_ELASTIC
  END IF

END SUBROUTINE MATERIAL_PROPERTIES_I
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
```

```
SUBROUTINE INTERNAL_PRESSURE_BCS

  USE VARIABLES
  IMPLICIT NONE

  j=0
  nbdy=r_nodes*NDF
  nbf=x_nodes

  IF(nbdy==0) THEN
    ibdy(1)=0
    vbdy(1)=0.D0
  ELSE
    nodbdy(1)=1;dofbdy(1)=1;vbdy(1)=0.D0 ! x, DOF have been re-ordered. Degree of freedom index: x=1, r=2, theta=3
    nodbdy(2)=1;dofbdy(2)=2;vbdy(2)=0.D0 ! r
    nodbdy(3)=1;dofbdy(3)=3;vbdy(3)=0.D0 ! theta
    DO i=NDF+1,nbdy,NDF
      j=j+1
      nodbdy(i  )=(j)*x_nodes+1;dofbdy(i  )=1;vbdy(i  )=0.D0 ! x, DOF have been re-ordered. Degree of freedom index: x=1, r=2, theta=3
      nodbdy(i+1)=(j)*x_nodes+1;dofbdy(i+1)=2;vbdy(i+1)=0.D0 ! r
      nodbdy(i+2)=(j)*x_nodes+1;dofbdy(i+2)=3;vbdy(i+2)=0.D0 ! theta
    END DO
    DO i=1,nbdy
      ibdy(i)=(nodbdy(i)-1)*ndf+dofbdy(i)
      !WRITE(*,*) nodbdy(i),dofbdy(i),vbdy(i),ibdy(i)
    END DO
  END IF

    CALL PRESSURE

 ! ###############################################################################################
 ! Constant internal Pressure  ! STATIC TRANSIENT SOLUTION (3/3)
 IF(CASE_FLAG==1) THEN
    IF(nbf==0) THEN
      ibf(1)=0
      vbf(1)=0.D0
    ELSE
      P_temp=P_constant*2.D0*pi*r_layer_loc(0)*x_length/x_elements ! P_constant p_interpolated(tt)
      nodbf(1)=1;dofbf(1)=2;vbf(1)=0.D0 ! P_temp/2.D0
      ibf(1)=(nodbf(1)-1)*ndf+dofbf(1)
      DO i=2,nbf-1
        nodbf(i)=i;dofbf(i)=2;vbf(i)=P_temp ! P_temp
      END DO
      nodbf(nbf)=nbf;dofbf(nbf)=2;vbf(nbf)=P_temp/2.D0 ! P_temp/2.D0
      DO i=1,nbf
        ibf(i)=(nodbf(i)-1)*ndf+dofbf(i)
        !WRITE(*,*) nodbf(i),dofbf(i),vbf(i),ibf(i)
      END DO
```

```
    END IF
  END IF
  ! ###############################################################################################


END SUBROUTINE INTERNAL_PRESSURE_BCS
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE VARIABLE_PRESSURE_BCS

  USE VARIABLES
  IMPLICIT NONE


  ! Variable internal Pressure
  IF(nbf==0) THEN
    ibf(1)=0
    vbf(1)=0.D0
  ELSE
    nodbf(1)=1;dofbf(1)=2;vbf(1)=0.D0 ! P_temp/2.D0
    ibf(1)=(nodbf(1)-1)*ndf+dofbf(1)
    DO i=2,nbf-1
      nodbf(i)=i;dofbf(i)=2;vbf(i)=P_variable(i,tt)*2.D0*pi*r_layer_loc(0)*x_length/x_elements ! P_temp
    END DO
    nodbf(nbf)=nbf;dofbf(nbf)=2;vbf(nbf)=P_variable(nbf,tt)*pi*r_layer_loc(0)*x_length/x_elements ! P_temp/2.D0
    DO i=1,nbf
      ibf(i)=(nodbf(i)-1)*ndf+dofbf(i)
      !WRITE(*,*) nodbf(i),dofbf(i),vbf(i),ibf(i)
    END DO
  END IF

    ! => nbf=x_nodes, P_variable(x_nodes,0:n_timesteps)


END SUBROUTINE VARIABLE_PRESSURE_BCS
!-----------------------------------------------------------------------------------------------------------------------------------!


!-----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE PRESSURE
  USE VARIABLES
  IMPLICIT NONE


  ! normal_p_vs_t(632)
  DO i=0,n_timesteps ! interpolate pressure from normal ballistics curve based on max pressure and time
```

```
  DO j=1,631
    IF((t(i)/time>normal_time_p_vs_t(j)).AND.(t(i)/time<normal_time_p_vs_t(j+1)))THEN
      x1=normal_time_p_vs_t(j)
      x2=normal_time_p_vs_t(j+1)
      y1=normal_p_vs_t(j)
      y2=normal_p_vs_t(j+1)
      p_interpolated(i)=((y2-y1)/(x2-x1)*(t(i)/time-x1)+y1)*P_max
    ELSE IF(t(i)/time==normal_time_p_vs_t(j))THEN
      p_interpolated(i)=normal_p_vs_t(j)*P_max
    END IF
  END DO
END DO


DO i=1,n_timesteps-1
  IF((p_interpolated(i)>p_interpolated(i-1)).and.(p_interpolated(i)>p_interpolated(i+1)))THEN
    interpolated_max=i ! p_interpolated(i)
  END IF
END DO


! normal_p_vs_x(658)
DO i=0,interpolated_max ! interpolate velocity from normal ballistics curve based on max pressure and time
  DO j=1,47 ! 47 is where the maximum occurs
    IF((p_interpolated(i)/P_max>normal_p_vs_x(j)).and.(p_interpolated(i)/P_max<normal_p_vs_x(j+1)))THEN
      x1=normal_position_p_vs_x(j)
      x2=normal_position_p_vs_x(j+1)
      y1=normal_p_vs_x(j)
      y2=normal_p_vs_x(j+1)
      x_interpolated(i)=((x2-x1)/(y2-y1)*(p_interpolated(i)/P_max-y1)+x1)*x_length
    ELSE IF(p_interpolated(i)/P_max==normal_p_vs_x(j))THEN
      x_interpolated(i)=normal_p_vs_x(j)*x_length
    END IF
  END DO
END DO
DO i=interpolated_max,n_timesteps ! interpolate velocity from normal ballistics curve based on max pressure and time
  DO j=47,657
    IF((p_interpolated(i)/P_max<normal_p_vs_x(j)).and.(p_interpolated(i)/P_max>normal_p_vs_x(j+1)))THEN
      x1=normal_position_p_vs_x(j)
      x2=normal_position_p_vs_x(j+1)
      y1=normal_p_vs_x(j)
      y2=normal_p_vs_x(j+1)
      x_interpolated(i)=((x2-x1)/(y2-y1)*(p_interpolated(i)/P_max-y1)+x1)*x_length
    ELSE IF(p_interpolated(i)/P_max==normal_p_vs_x(j))THEN
      x_interpolated(i)=normal_p_vs_x(j)*x_length
    END IF
  END DO
END DO
```

```
  DO i=0,n_timesteps
    DO j=1,x_nodes
      IF(x_position(j,tt)<x_interpolated(i))THEN
        P_variable(j,i)=p_interpolated(i)
      END IF
      IF(i==n_timesteps) P_variable(j,i)=p_interpolated(i)
    END DO
  END DO



  DO i=0,n_timesteps
    WRITE(30,*) i,",",p_interpolated(i),",",x_interpolated(i)
  END DO


END SUBROUTINE PRESSURE
!---------------------------------------------------------------------------------------------------------------------------------!


!---------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE UNBAND

  USE VARIABLES
  IMPLICIT NONE

  temp_matrix5=0.D0

!$$$$$$ WRITE(11,*) "BANDED"
!$$$$$$ DO i=1,NEQ
!$$$$$$   WRITE(11,200) (temp_matrix3(i,j), j=1,hbw)
!$$$$$$ END DO

  counter=0
  DO i=1,NEQ
    DO j=1,hbw
      temp_matrix5(i,j+counter)=temp_matrix3(i,j)
    END DO
    counter=counter+1
  END DO

  DO i=1,NEQ
    DO j=1,NEQ
      temp_matrix4(i,j)=temp_matrix5(i,j)
    END DO
  END DO
```

```
  200 FORMAT (5000(EN12.3,","))

!$$$$$$ WRITE(11,*) "UNBANDED - UPPER TRIANGULAR"
!$$$$$$ DO i=1,NEQ
!$$$$$$   WRITE(11,200) (temp_matrix4(i,j), j=1,NEQ)
!$$$$$$ END DO

  counter=1
  DO i=2,NEQ
    DO j=1,counter
      temp_matrix4(i,j)=temp_matrix5(j,i)
    END DO
    counter=counter+1
  END DO

!$$$$$$ WRITE(11,*)
!$$$$$$ WRITE(11,*) "UNBANDED - SYMMETRIC"
!$$$$$$ DO i=1,NEQ
!$$$$$$   WRITE(11,200) (temp_matrix4(i,j), j=1,NEQ)
!$$$$$$ END DO



END SUBROUTINE UNBAND
!----------------------------------------------------------------------------------------------------------------------------------!


!----------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE INVERSE
! the original matrix a(n,n) will be destroyed during the calculation

  USE VARIABLES
  IMPLICIT NONE



! step 0: initialization for matrices L and U and b

LL=0.D0
UU=0.D0
bb=0.D0

! step 1: forward elimination
do k=1, NEQ-1
   do i=k+1,NEQ
      coeff=aa(i,k)/aa(k,k)
```

```
      LL(i,k) = coeff
      do j=k+1,NEQ
         aa(i,j) = aa(i,j)-coeff*aa(k,j)
      end do
   end do
end do

! Step 2: prepare L and U matrices
! L matrix is a matrix of the elimination coefficient
! + the diagonal elements are 1.0
do i=1,NEQ
  LL(i,i) = 1.D0
end do
! U matrix is the upper triangular part of A
do j=1,NEQ
  do i=1,j
    UU(i,j) = aa(i,j)
  end do
end do

! Step 3: compute columns of the inverse matrix C
do k=1,NEQ
  bb(k)=1.D0
  dd(1) = bb(1)
! Step 3a: Solve Ld=b using the forward substitution
  do i=2,NEQ
    dd(i)=bb(i)
    do j=1,i-1
      dd(i) = dd(i) - LL(i,j)*dd(j)
    end do
  end do
! Step 3b: Solve Ux=d using the back substitution
  xx(NEQ)=dd(NEQ)/UU(NEQ,NEQ)
  do i = NEQ-1,1,-1
    xx(i) = dd(i)
    do j=NEQ,i+1,-1
      xx(i)=xx(i)-UU(i,j)*xx(j)
    end do
    xx(i) = xx(i)/uu(i,i)
  end do
! Step 3c: fill the solutions x(n) into column k of C
  do i=1,NEQ
    cc(i,k) = xx(i)
  end do
  bb(k)=0.D0
end do
END SUBROUTINE INVERSE
!------------------------------------------------------------------------------------------------------------------------------------!
```

```
!--------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE WRITE_STUFF

  USE VARIABLES
  IMPLICIT NONE

  WRITE(*,*)
  WRITE(*,*) "END PROGRAM"
  WRITE(*,*)
  WRITE(*,*)"cpu time:", T2-T1,"seconds"
  WRITE(*,*)" or ...", (T2-T1)/60.D0,"minutes"
  WRITE(*,*)" or ...", (T2-T1)/3600.D0,"hours"
  WRITE(*,*)
  WRITE(*,*) T4-T3,"seconds to invert mass matrix"
  WRITE(*,*) T6-T5,"seconds to invert effective matrix"
  WRITE(*,*)
  WRITE(*,*) "Delta Time",delta_time


!$$$$$$ WRITE(10,*) "Herakovich Table 10.1 Coupled Response of T300/5208 Tubes, wi/Ri"
!$$$$$$ WRITE(10,*) "                 (Pi=10psi, Ri/h=300, Ri=30 in)                 "
!$$$$$$ WRITE(10,*) "-----------------------------------------------------------------"

DO tt=0,ttt
  i=0
  DO n=1,x_nodes ! group displacements according to DOF (n=1,x_nodes => INSIDE RADIUS)
    i=i+1
    IF(tt==0) WRITE(20,*)
x_position(n,tt)+x_displacement_static(n),",",r_displacement_static(n)/r_layer_loc(0),",",n!,",",t(tt),",",n!
r_position(n,tt)+r_displacement(n,tt)
    !WRITE(10,*) x_position(n,tt)+x_displacement(n,tt),",",r_displacement(n,tt)/r_layer_loc(0),",",t(tt),",",n
    IF (n==6) WRITE(10,*) t(tt),",",r_displacement(n,tt)/r_layer_loc(0)!,",",p_interpolated(tt)
    !IF (n==6) WRITE(10,*) t(tt),",",r_position(n,tt)!r_displacement(n,tt)/r_layer_loc(0)
    !WRITE(10,*) t(tt),",",x_position(n,tt),",",r_position(n,tt)!r_displacement(n,tt)/r_layer_loc(0)
    !IF (n==5) WRITE(10,*) n,",",t(tt),",",r_displacement(n,tt)/r_layer_loc(0)
    IF(i==x_nodes) THEN
      !WRITE(10,*)
      i=0
    END IF
  END DO
END DO
!$$$$$$
!$$$$$$ 190 FORMAT (5000(EN12.3,","))
!$$$$$$
!$$$$$$ temp_matrix4=MATMUL(square_mass,mass_inverse)
!$$$$$$
```

```
!$$$$$$ DO i=1,NEQ
!$$$$$$   WRITE(11,190) (temp_matrix4(i,j), j=1,NEQ)
!$$$$$$ END DO
!$$$$$$
!$$$$$$ WRITE(11,*)
!$$$$$$
!$$$$$$ DO i=1,NEQ
!$$$$$$   WRITE(11,190) (square_stiff(i,j), j=1,NEQ)
!$$$$$$ END DO


!$$$$$$
!$$$$$$   108 FORMAT (5000I2)
!$$$$$$   rep_unbanded_matrix=0
!$$$$$$     DO i=1,NEQ
!$$$$$$       DO j=1,NEQ
!$$$$$$         IF((temp_matrix4(i,j)>0.D0).OR.(temp_matrix4(i,j)<0.D0)) THEN
!$$$$$$           rep_unbanded_matrix(i,j)=1
!$$$$$$         ELSE
!$$$$$$           rep_unbanded_matrix(i,j)=0
!$$$$$$         END IF
!$$$$$$       END DO
!$$$$$$     END DO
!$$$$$$   DO i=1,NEQ
!$$$$$$     WRITE(8,108) (rep_unbanded_matrix(i,j), j=1,NEQ)
!$$$$$$   END DO

END SUBROUTINE WRITE_STUFF
!------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------!
! GRAPHITE POLYMER COMPOSITE PHYSICAL PROPERTIES, SI UNITS
SUBROUTINE GRAPHITE_POLYMER

  USE VARIABLES
  IMPLICIT NONE

  E1=155.D9 ! Pa
  E2=12.1D9 ! Pa
  E3=12.1D9 ! Pa
  v23=0.458D0
  v13=0.248D0
  v12=0.248D0
  G23=3.2D9 ! Pa
  G13=4.4D9 ! Pa
  G12=4.4D9 ! Pa
```

```
  density=700.D0 ! kg/m^3

  visco_loss_factor=0.D0

END SUBROUTINE GRAPHITE_POLYMER
!---------------------------------------------------------------------------------------------------------------------------------!


!---------------------------------------------------------------------------------------------------------------------------------!
! Fronk's GRAPHITE POLYMER COMPOSITE PHYSICAL PROPERTIES, SI UNITS
SUBROUTINE FRONK_GRAPHITE_POLYMER

  USE VARIABLES
  IMPLICIT NONE

  E1=155.D9 ! Pa
  E2=12.0D9 ! Pa
  E3=12.0D9 ! Pa
  v23=0.458D0
  v13=0.248D0
  v12=0.248D0
  G23=4.1152263374486D9 ! Pa
  G13=4.4D9 ! Pa
  G12=4.4D9 ! Pa

  density=700.D0 ! kg/m^3

  visco_loss_factor=0.D0

END SUBROUTINE FRONK_GRAPHITE_POLYMER
!---------------------------------------------------------------------------------------------------------------------------------!


!---------------------------------------------------------------------------------------------------------------------------------!
! T300/5208, SI UNITS
SUBROUTINE T300_5280

  USE VARIABLES
  IMPLICIT NONE

  E1=132.0D9 ! Pa (19.2D6 psi)
  E2=10.8D9  ! Pa (1.56D6 psi)
  E3=10.8D9  ! Pa (1.56D6 psi)
  v23=0.59D0
  v13=0.24D0
  v12=0.24D0
  G23=3.38D9 ! Pa (0.49D6 psi)
  G13=5.65D9 ! Pa (0.82D6 psi)
```

```
  G12=5.65D9 ! Pa (0.82D6 psi)

  density=1540.D0 ! kg/m^3

  visco_loss_factor=0.D0

END SUBROUTINE T300_5280
!------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------!
! Avery 1125, SI UNITS
SUBROUTINE VISCO_ELASTIC

  USE VARIABLES
  IMPLICIT NONE

  E1=2068427.2D0 ! Pa (300 psi)
  E2=2068427.2D0 ! Pa (300 psi)
  E3=2068427.2D0 ! Pa (300 psi)
  v23=0.49D0
  v13=0.49D0
  v12=0.49D0
  G23=694103.1D0 ! Pa (100.67 psi)
  G13=694103.1D0 ! Pa (100.67 psi)
  G12=694103.1D0 ! Pa (100.67 psi)

  density=83.18D0 ! kg/m^3

  visco_loss_factor=1.D0
  visco_loss_factor2=visco_loss_factor

END SUBROUTINE VISCO_ELASTIC
!------------------------------------------------------------------------------------------------------------------------------!


!------------------------------------------------------------------------------------------------------------------------------!
! Stainless Steel, SI UNITS
SUBROUTINE STAINLESS_STEEL_SI

  USE VARIABLES
  IMPLICIT NONE

  E1=196.5D9 ! Pa (28.5D6 psi)
  E2=196.5D9 ! Pa
  E3=196.5D9 ! Pa
  v23=0.27D0
  v13=0.27D0
```

```
  v12=0.27D0
  G23=77.362204724409D9 ! Pa (11.220472440944881889763779527559 psi)
  G13=77.362204724409D9 ! Pa
  G12=77.362204724409D9 ! Pa

  density=2700.D0 ! kg/m^3

  visco_loss_factor=0.D0

END SUBROUTINE STAINLESS_STEEL_SI
!---------------------------------------------------------------------------------------------------------------------------------!


!---------------------------------------------------------------------------------------------------------------------------------!
! Stainless Steel, English UNITS
SUBROUTINE STAINLESS_STEEL_ENG

  USE VARIABLES
  IMPLICIT NONE

  E1=28.5D6 ! psi (196.5D9 Pa)
  E2=28.5D6 ! psi
  E3=128.5D6 ! psi
  v23=0.27D0
  v13=0.27D0
  v12=0.27D0
  G23=11.220472440944881889763779527559D6 ! psi (77.362204724409D9 Pa)
  G13=11.220472440944881889763779527559D6 ! psi
  G12=11.220472440944881889763779527559D6 ! psi

  density=270.D0 ! slug something /in^3

  visco_loss_factor=0.D0

END SUBROUTINE STAINLESS_STEEL_ENG
!---------------------------------------------------------------------------------------------------------------------------------!


!---------------------------------------------------------------------------------------------------------------------------------!
SUBROUTINE PROFILE
  USE VARIABLES
  IMPLICIT NONE


  !-----------------------------------------------
  normal_p_vs_x(1)=0.00000000000000D0
  normal_p_vs_x(2)=4.90545249670610D-02
  normal_p_vs_x(3)=1.42635522098082D-01
```

```
normal_p_vs_x(4)=2.39235221097837D-01
normal_p_vs_x(5)=3.32816218228858D-01
normal_p_vs_x(6)=4.29415918896447D-01
normal_p_vs_x(7)=4.71678960909324D-01
normal_p_vs_x(8)=5.16960706458768D-01
normal_p_vs_x(9)=5.80354673227617D-01
normal_p_vs_x(10)=6.04505499024590D-01
normal_p_vs_x(11)=6.49787244574034D-01
normal_p_vs_x(12)=6.73938070371008D-01
normal_p_vs_x(13)=7.13182410515151D-01
normal_p_vs_x(14)=7.34314531107724D-01
normal_p_vs_x(15)=7.52427949831562D-01
normal_p_vs_x(16)=7.73560072091969D-01
normal_p_vs_x(17)=7.94692194352375D-01
normal_p_vs_x(18)=8.15824314944947D-01
normal_p_vs_x(19)=8.27900328263485D-01
normal_p_vs_x(20)=8.49032448856057D-01
normal_p_vs_x(21)=8.58089758638028D-01
normal_p_vs_x(22)=8.70165768620898D-01
normal_p_vs_x(23)=8.82241783607269D-01
normal_p_vs_x(24)=8.94317795257974D-01
normal_p_vs_x(25)=9.03375105039944D-01
normal_p_vs_x(26)=9.09413711285347D-01
normal_p_vs_x(27)=9.18471019399484D-01
normal_p_vs_x(28)=9.24509625644886D-01
normal_p_vs_x(29)=9.33566935426857D-01
normal_p_vs_x(30)=9.39605541672260D-01
normal_p_vs_x(31)=9.45644144581996D-01
normal_p_vs_x(32)=9.54701456031799D-01
normal_p_vs_x(33)=9.57721358740635D-01
normal_p_vs_x(34)=9.63759964986038D-01
normal_p_vs_x(35)=9.69798571231441D-01
normal_p_vs_x(36)=9.72818473940276D-01
normal_p_vs_x(37)=9.75838376649112D-01
normal_p_vs_x(38)=9.78858279357947D-01
normal_p_vs_x(39)=9.81878182066783D-01
normal_p_vs_x(40)=9.84898086443452D-01
normal_p_vs_x(41)=9.87917989152288D-01
normal_p_vs_x(42)=9.87919188324556D-01
normal_p_vs_x(43)=9.90939091033392D-01
normal_p_vs_x(44)=9.93958993742227D-01
normal_p_vs_x(45)=9.96978896451063D-01
normal_p_vs_x(46)=9.96980097291164D-01
normal_p_vs_x(47)=1.00000000000000D+00
normal_p_vs_x(48)=9.96982497303534D-01
normal_p_vs_x(49)=9.93964992939235D-01
normal_p_vs_x(50)=9.93966193779337D-01
normal_p_vs_x(51)=9.90948689415038D-01
```

```
normal_p_vs_x(52)=9.90949890255140D-01
normal_p_vs_x(53)=9.87932385890841D-01
normal_p_vs_x(54)=9.87933586730942D-01
normal_p_vs_x(55)=9.84916080698810D-01
normal_p_vs_x(56)=9.84917283206745D-01
normal_p_vs_x(57)=9.81899778842446D-01
normal_p_vs_x(58)=9.78882276145980D-01
normal_p_vs_x(59)=9.75864773449515D-01
normal_p_vs_x(60)=9.72847269085216D-01
normal_p_vs_x(61)=9.69829766388750D-01
normal_p_vs_x(62)=9.66812262024451D-01
normal_p_vs_x(63)=9.63794759327985D-01
normal_p_vs_x(64)=9.63795956832420D-01
normal_p_vs_x(65)=9.60778455803788D-01
normal_p_vs_x(66)=9.57760953107322D-01
normal_p_vs_x(67)=9.54743448743023D-01
normal_p_vs_x(68)=9.51725946046558D-01
normal_p_vs_x(69)=9.45689739813525D-01
normal_p_vs_x(70)=9.45690938985793D-01
normal_p_vs_x(71)=9.45692139825895D-01
normal_p_vs_x(72)=9.36637230056295D-01
normal_p_vs_x(73)=9.30601023823262D-01
normal_p_vs_x(74)=9.27583519458963D-01
normal_p_vs_x(75)=9.24566016762497D-01
normal_p_vs_x(76)=9.21548512398198D-01
normal_p_vs_x(77)=9.18531009701732D-01
normal_p_vs_x(78)=9.15513507005267D-01
normal_p_vs_x(79)=9.06458597235666D-01
normal_p_vs_x(80)=9.03441092871367D-01
normal_p_vs_x(81)=9.00423590174902D-01
normal_p_vs_x(82)=8.97406087478436D-01
normal_p_vs_x(83)=8.91369881245403D-01
normal_p_vs_x(84)=8.88352376881104D-01
normal_p_vs_x(85)=8.85334874184639D-01
normal_p_vs_x(86)=8.82317369820339D-01
normal_p_vs_x(87)=8.76281163587307D-01
normal_p_vs_x(88)=8.73263660890841D-01
normal_p_vs_x(89)=8.70246158194375D-01
normal_p_vs_x(90)=8.67228653830076D-01
normal_p_vs_x(91)=8.64211151133611D-01
normal_p_vs_x(92)=8.61193646769312D-01
normal_p_vs_x(93)=8.55157440536279D-01
normal_p_vs_x(94)=8.52139937839813D-01
normal_p_vs_x(95)=8.46103731606780D-01
normal_p_vs_x(96)=8.40067525373747D-01
normal_p_vs_x(97)=8.37050021009448D-01
normal_p_vs_x(98)=8.34032518312983D-01
normal_p_vs_x(99)=8.27996312079950D-01
```

```
normal_p_vs_x(100)=8.24978807715651D-01
normal_p_vs_x(101)=8.21961305019185D-01
normal_p_vs_x(102)=8.18943802322719D-01
normal_p_vs_x(103)=8.15926297958420D-01
normal_p_vs_x(104)=8.12908795261955D-01
normal_p_vs_x(105)=8.09891292565489D-01
normal_p_vs_x(106)=8.06873788201190D-01
normal_p_vs_x(107)=8.00837581968157D-01
normal_p_vs_x(108)=7.97820079271691D-01
normal_p_vs_x(109)=7.94802574907392D-01
normal_p_vs_x(110)=7.88766368674359D-01
normal_p_vs_x(111)=7.85748865977894D-01
normal_p_vs_x(112)=7.82731361613595D-01
normal_p_vs_x(113)=7.79713858917129D-01
normal_p_vs_x(114)=7.76696356220663D-01
normal_p_vs_x(115)=7.70660149987631D-01
normal_p_vs_x(116)=7.67642645623332D-01
normal_p_vs_x(117)=7.64625142926866D-01
normal_p_vs_x(118)=7.61607638562567D-01
normal_p_vs_x(119)=7.55571432329534D-01
normal_p_vs_x(120)=7.52553929633068D-01
normal_p_vs_x(121)=7.49536426936603D-01
normal_p_vs_x(122)=7.49537626108871D-01
normal_p_vs_x(123)=7.46520123412405D-01
normal_p_vs_x(124)=7.40483917179372D-01
normal_p_vs_x(125)=7.37466412815073D-01
normal_p_vs_x(126)=7.34448910118608D-01
normal_p_vs_x(127)=7.31431405754309D-01
normal_p_vs_x(128)=7.25395199521276D-01
normal_p_vs_x(129)=7.22377696824810D-01
normal_p_vs_x(130)=7.19360194128345D-01
normal_p_vs_x(131)=7.16342689764046D-01
normal_p_vs_x(132)=7.13325187067580D-01
normal_p_vs_x(133)=7.10307682703281D-01
normal_p_vs_x(134)=7.10308883543383D-01
normal_p_vs_x(135)=7.01253973773782D-01
normal_p_vs_x(136)=6.98236471077317D-01
normal_p_vs_x(137)=6.95218966713018D-01
normal_p_vs_x(138)=6.92201464016552D-01
normal_p_vs_x(139)=6.89183959652253D-01
normal_p_vs_x(140)=6.86166456955787D-01
normal_p_vs_x(141)=6.83148954259322D-01
normal_p_vs_x(142)=6.83150153431590D-01
normal_p_vs_x(143)=6.80132650735124D-01
normal_p_vs_x(144)=6.77115146370825D-01
normal_p_vs_x(145)=6.74097643674360D-01
normal_p_vs_x(146)=6.68061437441327D-01
normal_p_vs_x(147)=6.65043933077028D-01
```

```
normal_p_vs_x(148)=6.65045133917129D-01
normal_p_vs_x(149)=6.62027629552830D-01
normal_p_vs_x(150)=6.59010126856365D-01
normal_p_vs_x(151)=6.55992624159899D-01
normal_p_vs_x(152)=6.52975119795600D-01
normal_p_vs_x(153)=6.49957617099134D-01
normal_p_vs_x(154)=6.46940114402669D-01
normal_p_vs_x(155)=6.43922610038370D-01
normal_p_vs_x(156)=6.40905107341904D-01
normal_p_vs_x(157)=6.40906306514172D-01
normal_p_vs_x(158)=6.37888803817707D-01
normal_p_vs_x(159)=6.34871299453408D-01
normal_p_vs_x(160)=6.31853796756942D-01
normal_p_vs_x(161)=6.28836294060477D-01
normal_p_vs_x(162)=6.25818789696178D-01
normal_p_vs_x(163)=6.22801285331879D-01
normal_p_vs_x(164)=6.19783784303246D-01
normal_p_vs_x(165)=6.16766279938947D-01
normal_p_vs_x(166)=6.16767480779049D-01
normal_p_vs_x(167)=6.13749976414750D-01
normal_p_vs_x(168)=6.10732473718284D-01
normal_p_vs_x(169)=6.07714969353985D-01
normal_p_vs_x(170)=6.04697466657520D-01
normal_p_vs_x(171)=6.01679963961054D-01
normal_p_vs_x(172)=6.01681163133322D-01
normal_p_vs_x(173)=5.98663660436857D-01
normal_p_vs_x(174)=5.95646156072558D-01
normal_p_vs_x(175)=5.92628653376092D-01
normal_p_vs_x(176)=5.89611150679626D-01
normal_p_vs_x(177)=5.86593646315327D-01
normal_p_vs_x(178)=5.83576143618862D-01
normal_p_vs_x(179)=5.83577342791130D-01
normal_p_vs_x(180)=5.80559840094664D-01
normal_p_vs_x(181)=5.77542335730365D-01
normal_p_vs_x(182)=5.74524833033900D-01
normal_p_vs_x(183)=5.71507330337434D-01
normal_p_vs_x(184)=5.71508529509702D-01
normal_p_vs_x(185)=5.68491026813237D-01
normal_p_vs_x(186)=5.65473522448938D-01
normal_p_vs_x(187)=5.65474721621206D-01
normal_p_vs_x(188)=5.62457217256907D-01
normal_p_vs_x(189)=5.59439716228275D-01
normal_p_vs_x(190)=5.59440915400543D-01
normal_p_vs_x(191)=5.56423412704077D-01
normal_p_vs_x(192)=5.53405910007612D-01
normal_p_vs_x(193)=5.50388405643313D-01
normal_p_vs_x(194)=5.47370902946847D-01
normal_p_vs_x(195)=5.44353400250381D-01
```

```
normal_p_vs_x(196)=5.44354599422650D-01
normal_p_vs_x(197)=5.41337096726184D-01
normal_p_vs_x(198)=5.38319592361885D-01
normal_p_vs_x(199)=5.38320793201987D-01
normal_p_vs_x(200)=5.35303288837688D-01
normal_p_vs_x(201)=5.32285786141222D-01
normal_p_vs_x(202)=5.32286985313490D-01
normal_p_vs_x(203)=5.29269482617025D-01
normal_p_vs_x(204)=5.26251978252726D-01
normal_p_vs_x(205)=5.23234475556260D-01
normal_p_vs_x(206)=5.23235674728528D-01
normal_p_vs_x(207)=5.20218172032063D-01
normal_p_vs_x(208)=5.17200669335597D-01
normal_p_vs_x(209)=5.17201868507865D-01
normal_p_vs_x(210)=5.17203067680134D-01
normal_p_vs_x(211)=5.14185564983668D-01
normal_p_vs_x(212)=5.11168062287203D-01
normal_p_vs_x(213)=5.11169259791637D-01
normal_p_vs_x(214)=5.08151758763005D-01
normal_p_vs_x(215)=5.08152957935273D-01
normal_p_vs_x(216)=5.05135455238808D-01
normal_p_vs_x(217)=5.02117950874509D-01
normal_p_vs_x(218)=4.99100448178043D-01
normal_p_vs_x(219)=4.99101647350311D-01
normal_p_vs_x(220)=4.96084144653846D-01
normal_p_vs_x(221)=4.96085343826114D-01
normal_p_vs_x(222)=4.93067841129648D-01
normal_p_vs_x(223)=4.93069040301917D-01
normal_p_vs_x(224)=4.90051537605451D-01
normal_p_vs_x(225)=4.87034034908986D-01
normal_p_vs_x(226)=4.84016530544686D-01
normal_p_vs_x(227)=4.84017731384788D-01
normal_p_vs_x(228)=4.81000227020489D-01
normal_p_vs_x(229)=4.81001427860591D-01
normal_p_vs_x(230)=4.81002627032859D-01
normal_p_vs_x(231)=4.77985124336393D-01
normal_p_vs_x(232)=4.74967619972094D-01
normal_p_vs_x(233)=4.74968820812196D-01
normal_p_vs_x(234)=4.71951316447897D-01
normal_p_vs_x(235)=4.68933813751431D-01
normal_p_vs_x(236)=4.68935012923700D-01
normal_p_vs_x(237)=4.65917510227234D-01
normal_p_vs_x(238)=4.65918709399502D-01
normal_p_vs_x(239)=4.62901206703037D-01
normal_p_vs_x(240)=4.62902405875305D-01
normal_p_vs_x(241)=4.59884903178839D-01
normal_p_vs_x(242)=4.59886100683274D-01
normal_p_vs_x(243)=4.56868599654642D-01
```

```
normal_p_vs_x(244)=4.56869798826910D-01
normal_p_vs_x(245)=4.53852296130445D-01
normal_p_vs_x(246)=4.53853495302713D-01
normal_p_vs_x(247)=4.50835992606247D-01
normal_p_vs_x(248)=4.47818489909782D-01
normal_p_vs_x(249)=4.47819689082050D-01
normal_p_vs_x(250)=4.44802186385584D-01
normal_p_vs_x(251)=4.44803385557853D-01
normal_p_vs_x(252)=4.41785882861387D-01
normal_p_vs_x(253)=4.41787082033655D-01
normal_p_vs_x(254)=4.38769577669356D-01
normal_p_vs_x(255)=4.38770778509458D-01
normal_p_vs_x(256)=4.35753275812992D-01
normal_p_vs_x(257)=4.35754474985261D-01
normal_p_vs_x(258)=4.32736972288795D-01
normal_p_vs_x(259)=4.32738171461063D-01
normal_p_vs_x(260)=4.29720668764598D-01
normal_p_vs_x(261)=4.29721867936866D-01
normal_p_vs_x(262)=4.29723067109134D-01
normal_p_vs_x(263)=4.26705564412669D-01
normal_p_vs_x(264)=4.26706763584937D-01
normal_p_vs_x(265)=4.23689260888471D-01
normal_p_vs_x(266)=4.23690460060740D-01
normal_p_vs_x(267)=4.20672957364274D-01
normal_p_vs_x(268)=4.20674156536542D-01
normal_p_vs_x(269)=4.20675357376644D-01
normal_p_vs_x(270)=4.17657853012345D-01
normal_p_vs_x(271)=4.17659053852447D-01
normal_p_vs_x(272)=4.14641549488148D-01
normal_p_vs_x(273)=4.14642750328249D-01
normal_p_vs_x(274)=4.11625245963950D-01
normal_p_vs_x(275)=4.11626446804052D-01
normal_p_vs_x(276)=4.11627645976320D-01
normal_p_vs_x(277)=4.08610143279855D-01
normal_p_vs_x(278)=4.08611342452123D-01
normal_p_vs_x(279)=4.05593839755657D-01
normal_p_vs_x(280)=4.02576335391358D-01
normal_p_vs_x(281)=4.02577536231460D-01
normal_p_vs_x(282)=4.02578735403728D-01
normal_p_vs_x(283)=3.99561232707263D-01
normal_p_vs_x(284)=3.99562431879531D-01
normal_p_vs_x(285)=3.99563631051799D-01
normal_p_vs_x(286)=3.96546128355333D-01
normal_p_vs_x(287)=3.96547327527602D-01
normal_p_vs_x(288)=3.93529824831136D-01
normal_p_vs_x(289)=3.93531024003404D-01
normal_p_vs_x(290)=3.90513521306939D-01
normal_p_vs_x(291)=3.90514718811374D-01
```

```
normal_p_vs_x(292)=3.90515921319309D-01
normal_p_vs_x(293)=3.87498416955010D-01
normal_p_vs_x(294)=3.87499617795111D-01
normal_p_vs_x(295)=3.87500815299546D-01
normal_p_vs_x(296)=3.84483314270914D-01
normal_p_vs_x(297)=3.84484513443182D-01
normal_p_vs_x(298)=3.81467010746717D-01
normal_p_vs_x(299)=3.81468209918985D-01
normal_p_vs_x(300)=3.78450707222519D-01
normal_p_vs_x(301)=3.78451906394788D-01
normal_p_vs_x(302)=3.78453107234889D-01
normal_p_vs_x(303)=3.78454306407158D-01
normal_p_vs_x(304)=3.75436803710692D-01
normal_p_vs_x(305)=3.72419297678559D-01
normal_p_vs_x(306)=3.72420500186495D-01
normal_p_vs_x(307)=3.72421699358763D-01
normal_p_vs_x(308)=3.69404196662297D-01
normal_p_vs_x(309)=3.69405394166732D-01
normal_p_vs_x(310)=3.69406595006834D-01
normal_p_vs_x(311)=3.66389092310368D-01
normal_p_vs_x(312)=3.66390291482636D-01
normal_p_vs_x(313)=3.66391492322738D-01
normal_p_vs_x(314)=3.63373987958439D-01
normal_p_vs_x(315)=3.63375188798541D-01
normal_p_vs_x(316)=3.63376387970809D-01
normal_p_vs_x(317)=3.60358885274344D-01
normal_p_vs_x(318)=3.60360084446612D-01
normal_p_vs_x(319)=3.57342581750146D-01
normal_p_vs_x(320)=3.57343780922414D-01
normal_p_vs_x(321)=3.57344981762516D-01
normal_p_vs_x(322)=3.54327477398217D-01
normal_p_vs_x(323)=3.54328678238319D-01
normal_p_vs_x(324)=3.54329877410587D-01
normal_p_vs_x(325)=3.51312374714122D-01
normal_p_vs_x(326)=3.51313573886390D-01
normal_p_vs_x(327)=3.51314773058658D-01
normal_p_vs_x(328)=3.48297270362192D-01
normal_p_vs_x(329)=3.48298469534461D-01
normal_p_vs_x(330)=3.45280966837995D-01
normal_p_vs_x(331)=3.45282166010263D-01
normal_p_vs_x(332)=3.45283366850365D-01
normal_p_vs_x(333)=3.45284566022633D-01
normal_p_vs_x(334)=3.45285763527068D-01
normal_p_vs_x(335)=3.42268262498436D-01
normal_p_vs_x(336)=3.42269461670704D-01
normal_p_vs_x(337)=3.42270662510806D-01
normal_p_vs_x(338)=3.39253158146507D-01
normal_p_vs_x(339)=3.39254358986609D-01
```

```
normal_p_vs_x(340)=3.36236854622310D-01
normal_p_vs_x(341)=3.36238055462411D-01
normal_p_vs_x(342)=3.36239254634679D-01
normal_p_vs_x(343)=3.36240455474781D-01
normal_p_vs_x(344)=3.33222951110482D-01
normal_p_vs_x(345)=3.33224151950584D-01
normal_p_vs_x(346)=3.33225351122852D-01
normal_p_vs_x(347)=3.33226550295120D-01
normal_p_vs_x(348)=3.30209047598655D-01
normal_p_vs_x(349)=3.30210246770923D-01
normal_p_vs_x(350)=3.27192744074457D-01
normal_p_vs_x(351)=3.27193943246726D-01
normal_p_vs_x(352)=3.27195144086827D-01
normal_p_vs_x(353)=3.27196343259096D-01
normal_p_vs_x(354)=3.24178840562630D-01
normal_p_vs_x(355)=3.24180039734898D-01
normal_p_vs_x(356)=3.21162535370599D-01
normal_p_vs_x(357)=3.21163736210701D-01
normal_p_vs_x(358)=3.21164937050803D-01
normal_p_vs_x(359)=3.21166136223071D-01
normal_p_vs_x(360)=3.21167335395339D-01
normal_p_vs_x(361)=3.18149832698874D-01
normal_p_vs_x(362)=3.18151031871142D-01
normal_p_vs_x(363)=3.18152232711244D-01
normal_p_vs_x(364)=3.18153431883512D-01
normal_p_vs_x(365)=3.15135929187046D-01
normal_p_vs_x(366)=3.15137128359315D-01
normal_p_vs_x(367)=3.15138327531583D-01
normal_p_vs_x(368)=3.12120824835117D-01
normal_p_vs_x(369)=3.12122024007385D-01
normal_p_vs_x(370)=3.12123224847487D-01
normal_p_vs_x(371)=3.09105720483188D-01
normal_p_vs_x(372)=3.09106921323290D-01
normal_p_vs_x(373)=3.09108120495558D-01
normal_p_vs_x(374)=3.09109321335660D-01
normal_p_vs_x(375)=3.09110520507928D-01
normal_p_vs_x(376)=3.06093017811462D-01
normal_p_vs_x(377)=3.06094216983731D-01
normal_p_vs_x(378)=3.06095414488166D-01
normal_p_vs_x(379)=3.03077913459533D-01
normal_p_vs_x(380)=3.03079112631802D-01
normal_p_vs_x(381)=3.03080313471903D-01
normal_p_vs_x(382)=3.00062809107604D-01
normal_p_vs_x(383)=3.00064009947706D-01
normal_p_vs_x(384)=3.00065209119974D-01
normal_p_vs_x(385)=3.00066408292243D-01
normal_p_vs_x(386)=3.00067609132344D-01
normal_p_vs_x(387)=2.97050103100212D-01
```

```
normal_p_vs_x(388)=2.97051305608147D-01
normal_p_vs_x(389)=2.97052504780415D-01
normal_p_vs_x(390)=2.97053703952683D-01
normal_p_vs_x(391)=2.94036199588384D-01
normal_p_vs_x(392)=2.94037402096320D-01
normal_p_vs_x(393)=2.94038601268588D-01
normal_p_vs_x(394)=2.94039800440856D-01
normal_p_vs_x(395)=2.91022296076557D-01
normal_p_vs_x(396)=2.91023496916659D-01
normal_p_vs_x(397)=2.91024697756760D-01
normal_p_vs_x(398)=2.88007193392461D-01
normal_p_vs_x(399)=2.88008394232563D-01
normal_p_vs_x(400)=2.88009593404831D-01
normal_p_vs_x(401)=2.88010794244933D-01
normal_p_vs_x(402)=2.88011993417201D-01
normal_p_vs_x(403)=2.88013192589470D-01
normal_p_vs_x(404)=2.88014393429571D-01
normal_p_vs_x(405)=2.84996889065272D-01
normal_p_vs_x(406)=2.84998089905374D-01
normal_p_vs_x(407)=2.84999289077642D-01
normal_p_vs_x(408)=2.85000488249911D-01
normal_p_vs_x(409)=2.81982985553445D-01
normal_p_vs_x(410)=2.81984184725713D-01
normal_p_vs_x(411)=2.81985385565815D-01
normal_p_vs_x(412)=2.78967881201516D-01
normal_p_vs_x(413)=2.78969082041618D-01
normal_p_vs_x(414)=2.78970281213886D-01
normal_p_vs_x(415)=2.78971482053988D-01
normal_p_vs_x(416)=2.75953977689688D-01
normal_p_vs_x(417)=2.75955178529790D-01
normal_p_vs_x(418)=2.75956377702058D-01
normal_p_vs_x(419)=2.75957576874327D-01
normal_p_vs_x(420)=2.75958777714429D-01
normal_p_vs_x(421)=2.75959976886697D-01
normal_p_vs_x(422)=2.75961176058965D-01
normal_p_vs_x(423)=2.72943673362499D-01
normal_p_vs_x(424)=2.72944872534768D-01
normal_p_vs_x(425)=2.72946073374869D-01
normal_p_vs_x(426)=2.72947272547138D-01
normal_p_vs_x(427)=2.69929769850672D-01
normal_p_vs_x(428)=2.69930969022940D-01
normal_p_vs_x(429)=2.69932169863042D-01
normal_p_vs_x(430)=2.69933369035310D-01
normal_p_vs_x(431)=2.66915866338845D-01
normal_p_vs_x(432)=2.66917065511113D-01
normal_p_vs_x(433)=2.66918264683381D-01
normal_p_vs_x(434)=2.66919465523483D-01
normal_p_vs_x(435)=2.66920664695751D-01
```

```
normal_p_vs_x(436)=2.63903161999286D-01
normal_p_vs_x(437)=2.63904361171554D-01
normal_p_vs_x(438)=2.63905560343822D-01
normal_p_vs_x(439)=2.63906761183924D-01
normal_p_vs_x(440)=2.63907960356192D-01
normal_p_vs_x(441)=2.63909161196294D-01
normal_p_vs_x(442)=2.63910360368562D-01
normal_p_vs_x(443)=2.60892856004263D-01
normal_p_vs_x(444)=2.60894056844365D-01
normal_p_vs_x(445)=2.60895256016633D-01
normal_p_vs_x(446)=2.60896456856735D-01
normal_p_vs_x(447)=2.57878952492436D-01
normal_p_vs_x(448)=2.57880153332537D-01
normal_p_vs_x(449)=2.57881352504806D-01
normal_p_vs_x(450)=2.57882553344907D-01
normal_p_vs_x(451)=2.54865048980608D-01
normal_p_vs_x(452)=2.54866249820710D-01
normal_p_vs_x(453)=2.54867448992978D-01
normal_p_vs_x(454)=2.54868646497413D-01
normal_p_vs_x(455)=2.54869849005348D-01
normal_p_vs_x(456)=2.51852344641049D-01
normal_p_vs_x(457)=2.51853545481151D-01
normal_p_vs_x(458)=2.51854742985586D-01
normal_p_vs_x(459)=2.51855943825687D-01
normal_p_vs_x(460)=2.51857144665789D-01
normal_p_vs_x(461)=2.51858343838057D-01
normal_p_vs_x(462)=2.51859544678159D-01
normal_p_vs_x(463)=2.48842040313860D-01
normal_p_vs_x(464)=2.48843241153962D-01
normal_p_vs_x(465)=2.48844440326230D-01
normal_p_vs_x(466)=2.48845639498498D-01
normal_p_vs_x(467)=2.45828136802033D-01
normal_p_vs_x(468)=2.45829335974301D-01
normal_p_vs_x(469)=2.45830536814403D-01
normal_p_vs_x(470)=2.45831735986671D-01
normal_p_vs_x(471)=2.45832935158939D-01
normal_p_vs_x(472)=2.42815432462474D-01
normal_p_vs_x(473)=2.42816629966908D-01
normal_p_vs_x(474)=2.42817832474844D-01
normal_p_vs_x(475)=2.42819031647112D-01
normal_p_vs_x(476)=2.42820232487214D-01
normal_p_vs_x(477)=2.42821431659482D-01
normal_p_vs_x(478)=2.42822630831750D-01
normal_p_vs_x(479)=2.42823831671852D-01
normal_p_vs_x(480)=2.39806327307553D-01
normal_p_vs_x(481)=2.39807528147654D-01
normal_p_vs_x(482)=2.39808727319923D-01
normal_p_vs_x(483)=2.39809924824358D-01
```

```
normal_p_vs_x(484)=2.36792423795725D-01
normal_p_vs_x(485)=2.36793622967994D-01
normal_p_vs_x(486)=2.36794823808095D-01
normal_p_vs_x(487)=2.36796021312530D-01
normal_p_vs_x(488)=2.36797223820465D-01
normal_p_vs_x(489)=2.33779719456166D-01
normal_p_vs_x(490)=2.30762216759701D-01
normal_p_vs_x(491)=2.30763415931969D-01
normal_p_vs_x(492)=2.30764616772071D-01
normal_p_vs_x(493)=2.30765815944339D-01
normal_p_vs_x(494)=2.30767015116607D-01
normal_p_vs_x(495)=2.30768214288875D-01
normal_p_vs_x(496)=2.30769415128977D-01
normal_p_vs_x(497)=2.30770615969079D-01
normal_p_vs_x(498)=2.30771815141347D-01
normal_p_vs_x(499)=2.30773014313615D-01
normal_p_vs_x(500)=2.30774215153717D-01
normal_p_vs_x(501)=2.27756710789418D-01
normal_p_vs_x(502)=2.27757911629520D-01
normal_p_vs_x(503)=2.27759110801788D-01
normal_p_vs_x(504)=2.27760309974056D-01
normal_p_vs_x(505)=2.27761510814158D-01
normal_p_vs_x(506)=2.24744006449859D-01
normal_p_vs_x(507)=2.24745207289961D-01
normal_p_vs_x(508)=2.24746406462229D-01
normal_p_vs_x(509)=2.24747605634497D-01
normal_p_vs_x(510)=2.24748806474599D-01
normal_p_vs_x(511)=2.24750005646867D-01
normal_p_vs_x(512)=2.21732502950402D-01
normal_p_vs_x(513)=2.21733702122670D-01
normal_p_vs_x(514)=2.21734902962772D-01
normal_p_vs_x(515)=2.21736102135040D-01
normal_p_vs_x(516)=2.21737299639475D-01
normal_p_vs_x(517)=2.21738502147410D-01
normal_p_vs_x(518)=2.18720997783111D-01
normal_p_vs_x(519)=2.18722198623212D-01
normal_p_vs_x(520)=2.18723396127647D-01
normal_p_vs_x(521)=2.18724598635582D-01
normal_p_vs_x(522)=2.18725797807851D-01
normal_p_vs_x(523)=2.18726996980119D-01
normal_p_vs_x(524)=2.18728197820221D-01
normal_p_vs_x(525)=2.18729396992489D-01
normal_p_vs_x(526)=2.18730594496924D-01
normal_p_vs_x(527)=2.18731797004859D-01
normal_p_vs_x(528)=2.15714292640560D-01
normal_p_vs_x(529)=2.15715493480662D-01
normal_p_vs_x(530)=2.15716690985096D-01
normal_p_vs_x(531)=2.15717893493032D-01
```

```
normal_p_vs_x(532)=2.15719092665300D-01
normal_p_vs_x(533)=2.15720291837568D-01
normal_p_vs_x(534)=2.12702787473269D-01
normal_p_vs_x(535)=2.12703988313371D-01
normal_p_vs_x(536)=2.12705189153473D-01
normal_p_vs_x(537)=2.12706388325741D-01
normal_p_vs_x(538)=2.12707589165842D-01
normal_p_vs_x(539)=2.12708788338111D-01
normal_p_vs_x(540)=2.09691285641645D-01
normal_p_vs_x(541)=2.09692484813913D-01
normal_p_vs_x(542)=2.09693683986182D-01
normal_p_vs_x(543)=2.09694884826283D-01
normal_p_vs_x(544)=2.09696082330718D-01
normal_p_vs_x(545)=2.09697283170820D-01
normal_p_vs_x(546)=2.06679780474354D-01
normal_p_vs_x(547)=2.06680979646623D-01
normal_p_vs_x(548)=2.06682178818891D-01
normal_p_vs_x(549)=2.06683379658993D-01
normal_p_vs_x(550)=2.06684580499094D-01
normal_p_vs_x(551)=2.06685779671363D-01
normal_p_vs_x(552)=2.06686978843631D-01
normal_p_vs_x(553)=2.06688179683733D-01
normal_p_vs_x(554)=2.06689377188167D-01
normal_p_vs_x(555)=2.06690578028269D-01
normal_p_vs_x(556)=2.03673075331803D-01
normal_p_vs_x(557)=2.03674274504072D-01
normal_p_vs_x(558)=2.03675473676340D-01
normal_p_vs_x(559)=2.03676674516442D-01
normal_p_vs_x(560)=2.00659171819976D-01
normal_p_vs_x(561)=2.00660370992244D-01
normal_p_vs_x(562)=2.00661570164513D-01
normal_p_vs_x(563)=2.00662771004614D-01
normal_p_vs_x(564)=2.00663970176883D-01
normal_p_vs_x(565)=2.00665171016984D-01
normal_p_vs_x(566)=2.00666370189253D-01
normal_p_vs_x(567)=2.00667569361521D-01
normal_p_vs_x(568)=2.00668768533789D-01
normal_p_vs_x(569)=2.00669969373891D-01
normal_p_vs_x(570)=1.97652466677425D-01
normal_p_vs_x(571)=1.97653665849693D-01
normal_p_vs_x(572)=1.97654865021962D-01
normal_p_vs_x(573)=1.97656065862063D-01
normal_p_vs_x(574)=1.97657265034332D-01
normal_p_vs_x(575)=1.97658465874433D-01
normal_p_vs_x(576)=1.97659665046702D-01
normal_p_vs_x(577)=1.97660864218970D-01
normal_p_vs_x(578)=1.97662065059072D-01
normal_p_vs_x(579)=1.97663264231340D-01
```

```
normal_p_vs_x(580)=1.97664465071442D-01
normal_p_vs_x(581)=1.97665664243710D-01
normal_p_vs_x(582)=1.97666863415978D-01
normal_p_vs_x(583)=1.94649360719513D-01
normal_p_vs_x(584)=1.94650559891781D-01
normal_p_vs_x(585)=1.94651760731883D-01
normal_p_vs_x(586)=1.94652959904151D-01
normal_p_vs_x(587)=1.94654160744253D-01
normal_p_vs_x(588)=1.94655359916521D-01
normal_p_vs_x(589)=1.94656559088789D-01
normal_p_vs_x(590)=1.94657759928891D-01
normal_p_vs_x(591)=1.91640255564592D-01
normal_p_vs_x(592)=1.91641456404693D-01
normal_p_vs_x(593)=1.91642655576962D-01
normal_p_vs_x(594)=1.91643854749230D-01
normal_p_vs_x(595)=1.91645055589332D-01
normal_p_vs_x(596)=1.91646254761600D-01
normal_p_vs_x(597)=1.91647455601702D-01
normal_p_vs_x(598)=1.88629951237403D-01
normal_p_vs_x(599)=1.88631152077504D-01
normal_p_vs_x(600)=1.88632351249773D-01
normal_p_vs_x(601)=1.88633550422041D-01
normal_p_vs_x(602)=1.88634751262143D-01
normal_p_vs_x(603)=1.88635950434411D-01
normal_p_vs_x(604)=1.88637149606679D-01
normal_p_vs_x(605)=1.85619646910213D-01
normal_p_vs_x(606)=1.85620846082482D-01
normal_p_vs_x(607)=1.85622046922583D-01
normal_p_vs_x(608)=1.85623246094852D-01
normal_p_vs_x(609)=1.85624446934953D-01
normal_p_vs_x(610)=1.85625644439388D-01
normal_p_vs_x(611)=1.85626845279490D-01
normal_p_vs_x(612)=1.85628046119592D-01
normal_p_vs_x(613)=1.85629245291860D-01
normal_p_vs_x(614)=1.85630444464128D-01
normal_p_vs_x(615)=1.85631645304230D-01
normal_p_vs_x(616)=1.85632844476498D-01
normal_p_vs_x(617)=1.85634045316600D-01
normal_p_vs_x(618)=1.85635244488868D-01
normal_p_vs_x(619)=1.82617741792403D-01
normal_p_vs_x(620)=1.82618939296837D-01
normal_p_vs_x(621)=1.82620140136939D-01
normal_p_vs_x(622)=1.82621340977041D-01
normal_p_vs_x(623)=1.82622540149309D-01
normal_p_vs_x(624)=1.82623739321577D-01
normal_p_vs_x(625)=1.82624940161679D-01
normal_p_vs_x(626)=1.82626137666114D-01
normal_p_vs_x(627)=1.79608636637482D-01
```

```
normal_p_vs_x(628)=1.79609835809750D-01
normal_p_vs_x(629)=1.79611036649852D-01
normal_p_vs_x(630)=1.79612235822120D-01
normal_p_vs_x(631)=1.79613434994388D-01
normal_p_vs_x(632)=1.79614635834490D-01
normal_p_vs_x(633)=1.79615835006758D-01
normal_p_vs_x(634)=1.76598330642459D-01
normal_p_vs_x(635)=1.76599531482561D-01
normal_p_vs_x(636)=1.76600732322663D-01
normal_p_vs_x(637)=1.76601931494931D-01
normal_p_vs_x(638)=1.76603130667199D-01
normal_p_vs_x(639)=1.76604331507301D-01
normal_p_vs_x(640)=1.76605529011736D-01
normal_p_vs_x(641)=1.76606729851837D-01
normal_p_vs_x(642)=1.76607930691939D-01
normal_p_vs_x(643)=1.73590426327640D-01
normal_p_vs_x(644)=1.73591627167742D-01
normal_p_vs_x(645)=1.73592826340010D-01
normal_p_vs_x(646)=1.73594027180112D-01
normal_p_vs_x(647)=1.73595226352380D-01
normal_p_vs_x(648)=1.73596425524648D-01
normal_p_vs_x(649)=1.73597626364750D-01
normal_p_vs_x(650)=1.73598823869185D-01
normal_p_vs_x(651)=1.73600024709287D-01
normal_p_vs_x(652)=1.73601225549388D-01
normal_p_vs_x(653)=1.73602424721656D-01
normal_p_vs_x(654)=1.73603625561758D-01
normal_p_vs_x(655)=1.73604824734026D-01
normal_p_vs_x(656)=1.73606023906295D-01
normal_p_vs_x(657)=1.73607223078563D-01
normal_p_vs_x(658)=1.70589720382097D-01
!-----------------------------------------------


!-----------------------------------------------
normal_p_vs_t(1)=6.43451922161886D-02
normal_p_vs_t(2)=6.58591969652661D-02
normal_p_vs_t(3)=6.73732017143437D-02
normal_p_vs_t(4)=6.83825382137287D-02
normal_p_vs_t(5)=6.93918747131137D-02
normal_p_vs_t(6)=7.04012112124987D-02
normal_p_vs_t(7)=7.34292207106538D-02
normal_p_vs_t(8)=7.49432246232438D-02
normal_p_vs_t(9)=7.64572285358338D-02
normal_p_vs_t(10)=7.79712332849113D-02
```

```
normal_p_vs_t(11)=7.94852380339889D-02
normal_p_vs_t(12)=8.09992427830664D-02
normal_p_vs_t(13)=8.25132475321439D-02
normal_p_vs_t(14)=8.40272522812215D-02
normal_p_vs_t(15)=8.55412570302990D-02
normal_p_vs_t(16)=8.85692648554790D-02
normal_p_vs_t(17)=8.95786013548641D-02
normal_p_vs_t(18)=9.05879378542491D-02
normal_p_vs_t(19)=9.15972743536341D-02
normal_p_vs_t(20)=9.46252838517891D-02
normal_p_vs_t(21)=9.61392886008667D-02
normal_p_vs_t(22)=9.76532933499442D-02
normal_p_vs_t(23)=1.00681301175124D-01
normal_p_vs_t(24)=1.02195305924202D-01
normal_p_vs_t(25)=1.03709310673279D-01
normal_p_vs_t(26)=1.06737320171434D-01
normal_p_vs_t(27)=1.09765329669589D-01
normal_p_vs_t(28)=1.11279333582179D-01
normal_p_vs_t(29)=1.12793337494769D-01
normal_p_vs_t(30)=1.15821346992925D-01
normal_p_vs_t(31)=1.17335351742002D-01
normal_p_vs_t(32)=1.18849356491080D-01
normal_p_vs_t(33)=1.21877365989235D-01
normal_p_vs_t(34)=1.23391369901825D-01
normal_p_vs_t(35)=1.24905373814415D-01
normal_p_vs_t(36)=1.27933383312570D-01
normal_p_vs_t(37)=1.30961392810725D-01
normal_p_vs_t(38)=1.33989402308880D-01
normal_p_vs_t(39)=1.40045419632215D-01
normal_p_vs_t(40)=1.43073429130370D-01
normal_p_vs_t(41)=1.46101438628525D-01
normal_p_vs_t(42)=1.49129446453705D-01
normal_p_vs_t(43)=1.52157455951860D-01
normal_p_vs_t(44)=1.55185465450015D-01
normal_p_vs_t(45)=1.58213474948170D-01
normal_p_vs_t(46)=1.61241482773350D-01
normal_p_vs_t(47)=1.67297501769660D-01
normal_p_vs_t(48)=1.79409538089306D-01
normal_p_vs_t(49)=1.85465557085616D-01
normal_p_vs_t(50)=1.88493564910796D-01
normal_p_vs_t(51)=1.91521574408951D-01
normal_p_vs_t(52)=1.94549583907106D-01
normal_p_vs_t(53)=2.00605601230441D-01
normal_p_vs_t(54)=2.03633610728596D-01
normal_p_vs_t(55)=2.06661620226751D-01
normal_p_vs_t(56)=2.12717637550086D-01
normal_p_vs_t(57)=2.15745647048241D-01
normal_p_vs_t(58)=2.18773656546396D-01
```

```
normal_p_vs_t(59)=2.24829673869731D-01
normal_p_vs_t(60)=2.27857683367886D-01
normal_p_vs_t(61)=2.30885692866042D-01
normal_p_vs_t(62)=2.36941710189377D-01
normal_p_vs_t(63)=2.39969719687532D-01
normal_p_vs_t(64)=2.42997729185687D-01
normal_p_vs_t(65)=2.46025738683842D-01
normal_p_vs_t(66)=2.52081756007177D-01
normal_p_vs_t(67)=2.55109765505332D-01
normal_p_vs_t(68)=2.64193792326822D-01
normal_p_vs_t(69)=2.67221801824977D-01
normal_p_vs_t(70)=2.70249811323132D-01
normal_p_vs_t(71)=2.76305828646467D-01
normal_p_vs_t(72)=2.82361847642777D-01
normal_p_vs_t(73)=2.85389855467957D-01
normal_p_vs_t(74)=2.91445874464268D-01
normal_p_vs_t(75)=2.97501891787603D-01
normal_p_vs_t(76)=3.00529901285758D-01
normal_p_vs_t(77)=3.06585920282068D-01
normal_p_vs_t(78)=3.12641937605403D-01
normal_p_vs_t(79)=3.18697956601713D-01
normal_p_vs_t(80)=3.21725964426893D-01
normal_p_vs_t(81)=3.27781983423203D-01
normal_p_vs_t(82)=3.30809992921358D-01
normal_p_vs_t(83)=3.39894019742848D-01
normal_p_vs_t(84)=3.42922029241004D-01
normal_p_vs_t(85)=3.48978046564339D-01
normal_p_vs_t(86)=3.52006056062494D-01
normal_p_vs_t(87)=3.61090082883984D-01
normal_p_vs_t(88)=3.67146101880294D-01
normal_p_vs_t(89)=3.73202119203629D-01
normal_p_vs_t(90)=3.76230128701784D-01
normal_p_vs_t(91)=3.85314155523274D-01
normal_p_vs_t(92)=3.91370174519584D-01
normal_p_vs_t(93)=3.97426191842919D-01
normal_p_vs_t(94)=4.00454201341075D-01
normal_p_vs_t(95)=4.09538228162565D-01
normal_p_vs_t(96)=4.15594247158875D-01
normal_p_vs_t(97)=4.21650264482210D-01
normal_p_vs_t(98)=4.24678273980365D-01
normal_p_vs_t(99)=4.33762300801855D-01
normal_p_vs_t(100)=4.39818319798165D-01
normal_p_vs_t(101)=4.45874335448525D-01
normal_p_vs_t(102)=4.51930356117810D-01
normal_p_vs_t(103)=4.57986371768170D-01
normal_p_vs_t(104)=4.64042392437456D-01
normal_p_vs_t(105)=4.67070400262636D-01
normal_p_vs_t(106)=4.76154428757101D-01
```

```
normal_p_vs_t(107)=4.82210444407461D-01
normal_p_vs_t(108)=4.88266465076746D-01
normal_p_vs_t(109)=4.94322480727106D-01
normal_p_vs_t(110)=5.03406509221571D-01
normal_p_vs_t(111)=5.09462528217881D-01
normal_p_vs_t(112)=5.18546555039372D-01
normal_p_vs_t(113)=5.24602574035682D-01
normal_p_vs_t(114)=5.30658591359017D-01
normal_p_vs_t(115)=5.39742618180507D-01
normal_p_vs_t(116)=5.45798637176817D-01
normal_p_vs_t(117)=5.51854654500152D-01
normal_p_vs_t(118)=5.60938682994617D-01
normal_p_vs_t(119)=5.66994700317952D-01
normal_p_vs_t(120)=5.70022709816107D-01
normal_p_vs_t(121)=5.82134746135753D-01
normal_p_vs_t(122)=5.88190763459088D-01
normal_p_vs_t(123)=5.94246782455398D-01
normal_p_vs_t(124)=6.09386828273198D-01
normal_p_vs_t(125)=6.15442845596533D-01
normal_p_vs_t(126)=6.21498864592843D-01
normal_p_vs_t(127)=6.27554881916178D-01
normal_p_vs_t(128)=6.33610900912489D-01
normal_p_vs_t(129)=6.39666918235824D-01
normal_p_vs_t(130)=6.45722937232134D-01
normal_p_vs_t(131)=6.54806964053624D-01
normal_p_vs_t(132)=6.60862981376959D-01
normal_p_vs_t(133)=6.66919000373269D-01
normal_p_vs_t(134)=6.72975017696604D-01
normal_p_vs_t(135)=6.82059046191069D-01
normal_p_vs_t(136)=6.88115063514405D-01
normal_p_vs_t(137)=6.94171082510715D-01
normal_p_vs_t(138)=7.00227099834050D-01
normal_p_vs_t(139)=7.09311126655540D-01
normal_p_vs_t(140)=7.15367145651850D-01
normal_p_vs_t(141)=7.21423162975185D-01
normal_p_vs_t(142)=7.27479181971495D-01
normal_p_vs_t(143)=7.36563208792985D-01
normal_p_vs_t(144)=7.42619227789295D-01
normal_p_vs_t(145)=7.48675245112631D-01
normal_p_vs_t(146)=7.54731264108941D-01
normal_p_vs_t(147)=7.60787281432276D-01
normal_p_vs_t(148)=7.66843300428586D-01
normal_p_vs_t(149)=7.72899317751921D-01
normal_p_vs_t(150)=7.81983344573411D-01
normal_p_vs_t(151)=7.88039363569721D-01
normal_p_vs_t(152)=7.94095380893056D-01
normal_p_vs_t(153)=7.97123390391211D-01
normal_p_vs_t(154)=8.06207417212702D-01
```

```
normal_p_vs_t(155)=8.12263436209012D-01
normal_p_vs_t(156)=8.18319453532347D-01
normal_p_vs_t(157)=8.21347463030502D-01
normal_p_vs_t(158)=8.27403482026812D-01
normal_p_vs_t(159)=8.30431489851992D-01
normal_p_vs_t(160)=8.36487508848302D-01
normal_p_vs_t(161)=8.42543527844612D-01
normal_p_vs_t(162)=8.48599545167947D-01
normal_p_vs_t(163)=8.51627554666102D-01
normal_p_vs_t(164)=8.54655564164257D-01
normal_p_vs_t(165)=8.63739590985748D-01
normal_p_vs_t(166)=8.66767600483903D-01
normal_p_vs_t(167)=8.72823617807238D-01
normal_p_vs_t(168)=8.75851627305393D-01
normal_p_vs_t(169)=8.81907644628728D-01
normal_p_vs_t(170)=8.87963663625038D-01
normal_p_vs_t(171)=8.90991673123193D-01
normal_p_vs_t(172)=8.94019680948373D-01
normal_p_vs_t(173)=9.00075699944683D-01
normal_p_vs_t(174)=9.06131717268018D-01
normal_p_vs_t(175)=9.09159726766173D-01
normal_p_vs_t(176)=9.10673731515251D-01
normal_p_vs_t(177)=9.12187736264328D-01
normal_p_vs_t(178)=9.15215745762484D-01
normal_p_vs_t(179)=9.21271763085819D-01
normal_p_vs_t(180)=9.27327782082129D-01
normal_p_vs_t(181)=9.30355789907309D-01
normal_p_vs_t(182)=9.33383799405464D-01
normal_p_vs_t(183)=9.36411808903619D-01
normal_p_vs_t(184)=9.39439818401774D-01
normal_p_vs_t(185)=9.42467826226954D-01
normal_p_vs_t(186)=9.45495835725109D-01
normal_p_vs_t(187)=9.51551854721419D-01
normal_p_vs_t(188)=9.54579862546599D-01
normal_p_vs_t(189)=9.60635881542909D-01
normal_p_vs_t(190)=9.69719908364399D-01
normal_p_vs_t(191)=9.71233913113477D-01
normal_p_vs_t(192)=9.72747917862554D-01
normal_p_vs_t(193)=9.75775927360710D-01
normal_p_vs_t(194)=9.78803935185890D-01
normal_p_vs_t(195)=9.79813271685275D-01
normal_p_vs_t(196)=9.80822608184660D-01
normal_p_vs_t(197)=9.81831944684045D-01
normal_p_vs_t(198)=9.84859954182200D-01
normal_p_vs_t(199)=9.86373958931277D-01
normal_p_vs_t(200)=9.87887963680355D-01
normal_p_vs_t(201)=9.90915971505535D-01
normal_p_vs_t(202)=9.91672973880074D-01
```

```
normal_p_vs_t(203)=9.92429976254612D-01
normal_p_vs_t(204)=9.93186978629151D-01
normal_p_vs_t(205)=9.93943981003690D-01
normal_p_vs_t(206)=9.95457985752767D-01
normal_p_vs_t(207)=9.96971990501845D-01
normal_p_vs_t(208)=1.00000000000000D+00
normal_p_vs_t(209)=9.96971990501845D-01
normal_p_vs_t(210)=9.95962654002460D-01
normal_p_vs_t(211)=9.94953317503075D-01
normal_p_vs_t(212)=9.93943981003690D-01
normal_p_vs_t(213)=9.93565479816420D-01
normal_p_vs_t(214)=9.93186978629151D-01
normal_p_vs_t(215)=9.92808477441882D-01
normal_p_vs_t(216)=9.92429976254612D-01
normal_p_vs_t(217)=9.92051475067343D-01
normal_p_vs_t(218)=9.91672973880074D-01
normal_p_vs_t(219)=9.91294472692804D-01
normal_p_vs_t(220)=9.90915971505535D-01
normal_p_vs_t(221)=9.89401967592945D-01
normal_p_vs_t(222)=9.87887963680355D-01
normal_p_vs_t(223)=9.86373958931277D-01
normal_p_vs_t(224)=9.84859954182200D-01
normal_p_vs_t(225)=9.83850617682815D-01
normal_p_vs_t(226)=9.82841281183430D-01
normal_p_vs_t(227)=9.81831944684045D-01
normal_p_vs_t(228)=9.78803935185890D-01
normal_p_vs_t(229)=9.77289931273300D-01
normal_p_vs_t(230)=9.75775927360710D-01
normal_p_vs_t(231)=9.72747917862554D-01
normal_p_vs_t(232)=9.71233913113477D-01
normal_p_vs_t(233)=9.69719908364399D-01
normal_p_vs_t(234)=9.66691898866244D-01
normal_p_vs_t(235)=9.63663891041064D-01
normal_p_vs_t(236)=9.62149886291987D-01
normal_p_vs_t(237)=9.60635881542909D-01
normal_p_vs_t(238)=9.57607872044754D-01
normal_p_vs_t(239)=9.54579862546599D-01
normal_p_vs_t(240)=9.51551854721419D-01
normal_p_vs_t(241)=9.50037849972342D-01
normal_p_vs_t(242)=9.48523845223264D-01
normal_p_vs_t(243)=9.45495835725109D-01
normal_p_vs_t(244)=9.42467826226954D-01
normal_p_vs_t(245)=9.39439818401774D-01
normal_p_vs_t(246)=9.36411808903619D-01
normal_p_vs_t(247)=9.33383799405464D-01
normal_p_vs_t(248)=9.30355789907309D-01
normal_p_vs_t(249)=9.27327782082129D-01
normal_p_vs_t(250)=9.24299772583974D-01
```

```
normal_p_vs_t(251)=9.21271763085819D-01
normal_p_vs_t(252)=9.18243753587664D-01
normal_p_vs_t(253)=9.15215745762484D-01
normal_p_vs_t(254)=9.09159726766173D-01
normal_p_vs_t(255)=9.00075699944683D-01
normal_p_vs_t(256)=8.97047690446528D-01
normal_p_vs_t(257)=8.94019680948373D-01
normal_p_vs_t(258)=8.90991673123193D-01
normal_p_vs_t(259)=8.87963663625038D-01
normal_p_vs_t(260)=8.84935654126883D-01
normal_p_vs_t(261)=8.78879636803548D-01
normal_p_vs_t(262)=8.75851627305393D-01
normal_p_vs_t(263)=8.72823617807238D-01
normal_p_vs_t(264)=8.69795608309083D-01
normal_p_vs_t(265)=8.63739590985748D-01
normal_p_vs_t(266)=8.60711581487593D-01
normal_p_vs_t(267)=8.57683571989437D-01
normal_p_vs_t(268)=8.54655564164257D-01
normal_p_vs_t(269)=8.48599545167947D-01
normal_p_vs_t(270)=8.45571535669792D-01
normal_p_vs_t(271)=8.42543527844612D-01
normal_p_vs_t(272)=8.39515518346457D-01
normal_p_vs_t(273)=8.33459499350147D-01
normal_p_vs_t(274)=8.30431489851992D-01
normal_p_vs_t(275)=8.27403482026812D-01
normal_p_vs_t(276)=8.24375472528657D-01
normal_p_vs_t(277)=8.18319453532347D-01
normal_p_vs_t(278)=8.15291445707167D-01
normal_p_vs_t(279)=8.12263436209012D-01
normal_p_vs_t(280)=8.06207417212702D-01
normal_p_vs_t(281)=8.04693413300112D-01
normal_p_vs_t(282)=8.03179409387522D-01
normal_p_vs_t(283)=7.97123390391211D-01
normal_p_vs_t(284)=7.94095380893056D-01
normal_p_vs_t(285)=7.88039363569721D-01
normal_p_vs_t(286)=7.85011354071566D-01
normal_p_vs_t(287)=7.81983344573411D-01
normal_p_vs_t(288)=7.78955336748231D-01
normal_p_vs_t(289)=7.72899317751921D-01
normal_p_vs_t(290)=7.69871308253766D-01
normal_p_vs_t(291)=7.66843300428586D-01
normal_p_vs_t(292)=7.63815290930431D-01
normal_p_vs_t(293)=7.57759271934121D-01
normal_p_vs_t(294)=7.54731264108941D-01
normal_p_vs_t(295)=7.48675245112631D-01
normal_p_vs_t(296)=7.45647235614476D-01
normal_p_vs_t(297)=7.39591218291140D-01
normal_p_vs_t(298)=7.36563208792985D-01
```

```
normal_p_vs_t(299)=7.33535199294830D-01
normal_p_vs_t(300)=7.30507191469650D-01
normal_p_vs_t(301)=7.24451172473340D-01
normal_p_vs_t(302)=7.21423162975185D-01
normal_p_vs_t(303)=7.18395153477030D-01
normal_p_vs_t(304)=7.15367145651850D-01
normal_p_vs_t(305)=7.09311126655540D-01
normal_p_vs_t(306)=7.06283117157385D-01
normal_p_vs_t(307)=7.03255109332205D-01
normal_p_vs_t(308)=7.00227099834050D-01
normal_p_vs_t(309)=6.94171082510715D-01
normal_p_vs_t(310)=6.91143073012560D-01
normal_p_vs_t(311)=6.89629068263482D-01
normal_p_vs_t(312)=6.88115063514405D-01
normal_p_vs_t(313)=6.82059046191069D-01
normal_p_vs_t(314)=6.76003027194759D-01
normal_p_vs_t(315)=6.72975017696604D-01
normal_p_vs_t(316)=6.69947009871424D-01
normal_p_vs_t(317)=6.63890990875114D-01
normal_p_vs_t(318)=6.60862981376959D-01
normal_p_vs_t(319)=6.57834973551779D-01
normal_p_vs_t(320)=6.54806964053624D-01
normal_p_vs_t(321)=6.51778954555469D-01
normal_p_vs_t(322)=6.39666918235824D-01
normal_p_vs_t(323)=6.36638908737669D-01
normal_p_vs_t(324)=6.33610900912489D-01
normal_p_vs_t(325)=6.30582891414333D-01
normal_p_vs_t(326)=6.27554881916178D-01
normal_p_vs_t(327)=6.24526872418023D-01
normal_p_vs_t(328)=6.21498864592843D-01
normal_p_vs_t(329)=6.12414836098378D-01
normal_p_vs_t(330)=6.09386828273198D-01
normal_p_vs_t(331)=6.03330809276888D-01
normal_p_vs_t(332)=6.00302799778733D-01
normal_p_vs_t(333)=5.97274791953553D-01
normal_p_vs_t(334)=5.88190763459088D-01
normal_p_vs_t(335)=5.85162755633908D-01
normal_p_vs_t(336)=5.82134746135753D-01
normal_p_vs_t(337)=5.79106736637598D-01
normal_p_vs_t(338)=5.76078727139443D-01
normal_p_vs_t(339)=5.66994700317952D-01
normal_p_vs_t(340)=5.65480695568875D-01
normal_p_vs_t(341)=5.63966690819797D-01
normal_p_vs_t(342)=5.60938682994617D-01
normal_p_vs_t(343)=5.57910673496462D-01
normal_p_vs_t(344)=5.48826646674972D-01
normal_p_vs_t(345)=5.45798637176817D-01
normal_p_vs_t(346)=5.44284632427740D-01
```

```
normal_p_vs_t(347)=5.42770627678662D-01
normal_p_vs_t(348)=5.39742618180507D-01
normal_p_vs_t(349)=5.36714610355327D-01
normal_p_vs_t(350)=5.33686600857172D-01
normal_p_vs_t(351)=5.27630581860862D-01
normal_p_vs_t(352)=5.21574564537527D-01
normal_p_vs_t(353)=5.20060559788449D-01
normal_p_vs_t(354)=5.18546555039372D-01
normal_p_vs_t(355)=5.15518545541216D-01
normal_p_vs_t(356)=5.12490537716036D-01
normal_p_vs_t(357)=5.09462528217881D-01
normal_p_vs_t(358)=5.06434518719726D-01
normal_p_vs_t(359)=5.04920513970649D-01
normal_p_vs_t(360)=5.03406509221571D-01
normal_p_vs_t(361)=5.00378501396391D-01
normal_p_vs_t(362)=4.97350491898236D-01
normal_p_vs_t(363)=4.94322480727106D-01
normal_p_vs_t(364)=4.91294472901926D-01
normal_p_vs_t(365)=4.88266465076746D-01
normal_p_vs_t(366)=4.85238455578591D-01
normal_p_vs_t(367)=4.82210444407461D-01
normal_p_vs_t(368)=4.80696440494871D-01
normal_p_vs_t(369)=4.79182436582281D-01
normal_p_vs_t(370)=4.76154428757101D-01
normal_p_vs_t(371)=4.73126419258946D-01
normal_p_vs_t(372)=4.71612413673381D-01
normal_p_vs_t(373)=4.70098408087816D-01
normal_p_vs_t(374)=4.67070400262636D-01
normal_p_vs_t(375)=4.64042392437456D-01
normal_p_vs_t(376)=4.62528387688378D-01
normal_p_vs_t(377)=4.61014382939301D-01
normal_p_vs_t(378)=4.57986371768170D-01
normal_p_vs_t(379)=4.56472367855580D-01
normal_p_vs_t(380)=4.54958363942990D-01
normal_p_vs_t(381)=4.51930356117810D-01
normal_p_vs_t(382)=4.48902346619655D-01
normal_p_vs_t(383)=4.45874335448525D-01
normal_p_vs_t(384)=4.44360331535935D-01
normal_p_vs_t(385)=4.42846327623345D-01
normal_p_vs_t(386)=4.39818319798165D-01
normal_p_vs_t(387)=4.38304315049088D-01
normal_p_vs_t(388)=4.36790310300010D-01
normal_p_vs_t(389)=4.35276305550933D-01
normal_p_vs_t(390)=4.33762300801855D-01
normal_p_vs_t(391)=4.30734291303700D-01
normal_p_vs_t(392)=4.29220287391110D-01
normal_p_vs_t(393)=4.27706283478520D-01
normal_p_vs_t(394)=4.24678273980365D-01
```

```
normal_p_vs_t(395)=4.21650264482210D-01
normal_p_vs_t(396)=4.20136259733132D-01
normal_p_vs_t(397)=4.18622254984055D-01
normal_p_vs_t(398)=4.17108251071465D-01
normal_p_vs_t(399)=4.15594247158875D-01
normal_p_vs_t(400)=4.12566237660720D-01
normal_p_vs_t(401)=4.11052232911642D-01
normal_p_vs_t(402)=4.09538228162565D-01
normal_p_vs_t(403)=4.08024223413487D-01
normal_p_vs_t(404)=4.06510218664410D-01
normal_p_vs_t(405)=4.03482210839230D-01
normal_p_vs_t(406)=4.00454201341075D-01
normal_p_vs_t(407)=3.99444864841690D-01
normal_p_vs_t(408)=3.98435528342304D-01
normal_p_vs_t(409)=3.97426191842919D-01
normal_p_vs_t(410)=3.94398182344764D-01
normal_p_vs_t(411)=3.92884178432174D-01
normal_p_vs_t(412)=3.91370174519584D-01
normal_p_vs_t(413)=3.88342165021429D-01
normal_p_vs_t(414)=3.86828160272352D-01
normal_p_vs_t(415)=3.85314155523274D-01
normal_p_vs_t(416)=3.83800150774197D-01
normal_p_vs_t(417)=3.82286146025119D-01
normal_p_vs_t(418)=3.79258138199939D-01
normal_p_vs_t(419)=3.78248801700554D-01
normal_p_vs_t(420)=3.77239465201169D-01
normal_p_vs_t(421)=3.76230128701784D-01
normal_p_vs_t(422)=3.73202119203629D-01
normal_p_vs_t(423)=3.70174109705474D-01
normal_p_vs_t(424)=3.68660105792884D-01
normal_p_vs_t(425)=3.67146101880294D-01
normal_p_vs_t(426)=3.65632097131216D-01
normal_p_vs_t(427)=3.64118092382139D-01
normal_p_vs_t(428)=3.62604087633061D-01
normal_p_vs_t(429)=3.61090082883984D-01
normal_p_vs_t(430)=3.58062073385829D-01
normal_p_vs_t(431)=3.56548069473239D-01
normal_p_vs_t(432)=3.55034065560649D-01
normal_p_vs_t(433)=3.53520060811571D-01
normal_p_vs_t(434)=3.52006056062494D-01
normal_p_vs_t(435)=3.50492051313416D-01
normal_p_vs_t(436)=3.48978046564339D-01
normal_p_vs_t(437)=3.47464041815261D-01
normal_p_vs_t(438)=3.45950037066184D-01
normal_p_vs_t(439)=3.44940701124457D-01
normal_p_vs_t(440)=3.43931365182730D-01
normal_p_vs_t(441)=3.42922029241004D-01
normal_p_vs_t(442)=3.41408024491926D-01
```

```
normal_p_vs_t(443)=3.39894019742848D-01
normal_p_vs_t(444)=3.38380014993771D-01
normal_p_vs_t(445)=3.36866010244693D-01
normal_p_vs_t(446)=3.33838000746538D-01
normal_p_vs_t(447)=3.32828664804812D-01
normal_p_vs_t(448)=3.31819328863085D-01
normal_p_vs_t(449)=3.30809992921358D-01
normal_p_vs_t(450)=3.29800656421973D-01
normal_p_vs_t(451)=3.28791319922588D-01
normal_p_vs_t(452)=3.27781983423203D-01
normal_p_vs_t(453)=3.24753973925048D-01
normal_p_vs_t(454)=3.23239969175971D-01
normal_p_vs_t(455)=3.21725964426893D-01
normal_p_vs_t(456)=3.20968962470598D-01
normal_p_vs_t(457)=3.20211960514303D-01
normal_p_vs_t(458)=3.19454958558008D-01
normal_p_vs_t(459)=3.18697956601713D-01
normal_p_vs_t(460)=3.15669947103558D-01
normal_p_vs_t(461)=3.14155942354481D-01
normal_p_vs_t(462)=3.12641937605403D-01
normal_p_vs_t(463)=3.11127932856325D-01
normal_p_vs_t(464)=3.09613928107248D-01
normal_p_vs_t(465)=3.08604592165521D-01
normal_p_vs_t(466)=3.07595256223795D-01
normal_p_vs_t(467)=3.06585920282068D-01
normal_p_vs_t(468)=3.05071915532990D-01
normal_p_vs_t(469)=3.03557910783913D-01
normal_p_vs_t(470)=3.02043906034835D-01
normal_p_vs_t(471)=3.00529901285758D-01
normal_p_vs_t(472)=2.99520564786373D-01
normal_p_vs_t(473)=2.98511228286988D-01
normal_p_vs_t(474)=2.97501891787603D-01
normal_p_vs_t(475)=2.96492555845876D-01
normal_p_vs_t(476)=2.95483219904149D-01
normal_p_vs_t(477)=2.94473883962423D-01
normal_p_vs_t(478)=2.92959879213345D-01
normal_p_vs_t(479)=2.91445874464268D-01
normal_p_vs_t(480)=2.89931869715190D-01
normal_p_vs_t(481)=2.88417864966113D-01
normal_p_vs_t(482)=2.87913196716420D-01
normal_p_vs_t(483)=2.87408528466727D-01
normal_p_vs_t(484)=2.86903860217035D-01
normal_p_vs_t(485)=2.86399191967342D-01
normal_p_vs_t(486)=2.85894523717650D-01
normal_p_vs_t(487)=2.85389855467957D-01
normal_p_vs_t(488)=2.82361847642777D-01
normal_p_vs_t(489)=2.81352511143392D-01
normal_p_vs_t(490)=2.80343174644007D-01
```

```
normal_p_vs_t(491)=2.79333838144622D-01
normal_p_vs_t(492)=2.76305828646467D-01
normal_p_vs_t(493)=2.75700226746836D-01
normal_p_vs_t(494)=2.75094624847205D-01
normal_p_vs_t(495)=2.74489022947574D-01
normal_p_vs_t(496)=2.73883421047943D-01
normal_p_vs_t(497)=2.73277819148312D-01
normal_p_vs_t(498)=2.71763815235722D-01
normal_p_vs_t(499)=2.70249811323132D-01
normal_p_vs_t(500)=2.69240474823747D-01
normal_p_vs_t(501)=2.68231138324362D-01
normal_p_vs_t(502)=2.67221801824977D-01
normal_p_vs_t(503)=2.65707797075900D-01
normal_p_vs_t(504)=2.64193792326822D-01
normal_p_vs_t(505)=2.63436789952283D-01
normal_p_vs_t(506)=2.62679787577745D-01
normal_p_vs_t(507)=2.61922785203206D-01
normal_p_vs_t(508)=2.61165782828667D-01
normal_p_vs_t(509)=2.60156446886940D-01
normal_p_vs_t(510)=2.59147110945214D-01
normal_p_vs_t(511)=2.58137775003487D-01
normal_p_vs_t(512)=2.57128438504102D-01
normal_p_vs_t(513)=2.56119102004717D-01
normal_p_vs_t(514)=2.55109765505332D-01
normal_p_vs_t(515)=2.53595760756254D-01
normal_p_vs_t(516)=2.52081756007177D-01
normal_p_vs_t(517)=2.51324753632638D-01
normal_p_vs_t(518)=2.50567751258099D-01
normal_p_vs_t(519)=2.49810748883561D-01
normal_p_vs_t(520)=2.49053746509022D-01
normal_p_vs_t(521)=2.48044410567295D-01
normal_p_vs_t(522)=2.47035074625568D-01
normal_p_vs_t(523)=2.46025738683842D-01
normal_p_vs_t(524)=2.45016402184457D-01
normal_p_vs_t(525)=2.44007065685072D-01
normal_p_vs_t(526)=2.42997729185687D-01
normal_p_vs_t(527)=2.42240726811148D-01
normal_p_vs_t(528)=2.41483724436609D-01
normal_p_vs_t(529)=2.40726722062070D-01
normal_p_vs_t(530)=2.39969719687532D-01
normal_p_vs_t(531)=2.38960383188147D-01
normal_p_vs_t(532)=2.37951046688762D-01
normal_p_vs_t(533)=2.36941710189377D-01
normal_p_vs_t(534)=2.35932374247650D-01
normal_p_vs_t(535)=2.34923038305923D-01
normal_p_vs_t(536)=2.33913702364197D-01
normal_p_vs_t(537)=2.32904365864812D-01
normal_p_vs_t(538)=2.31895029365427D-01
```

```
normal_p_vs_t(539)=2.30885692866042D-01
normal_p_vs_t(540)=2.30280090966411D-01
normal_p_vs_t(541)=2.29674489066780D-01
normal_p_vs_t(542)=2.29068887167148D-01
normal_p_vs_t(543)=2.28463285267517D-01
normal_p_vs_t(544)=2.27857683367886D-01
normal_p_vs_t(545)=2.26848346868501D-01
normal_p_vs_t(546)=2.25839010369116D-01
normal_p_vs_t(547)=2.24829673869731D-01
normal_p_vs_t(548)=2.23820337928005D-01
normal_p_vs_t(549)=2.22811001986278D-01
normal_p_vs_t(550)=2.21801666044551D-01
normal_p_vs_t(551)=2.20792329545166D-01
normal_p_vs_t(552)=2.19782993045781D-01
normal_p_vs_t(553)=2.18773656546396D-01
normal_p_vs_t(554)=2.18268988296704D-01
normal_p_vs_t(555)=2.17764320047011D-01
normal_p_vs_t(556)=2.17259651797319D-01
normal_p_vs_t(557)=2.16754983547626D-01
normal_p_vs_t(558)=2.16250315297934D-01
normal_p_vs_t(559)=2.15745647048241D-01
normal_p_vs_t(560)=2.14736310548856D-01
normal_p_vs_t(561)=2.13726974049471D-01
normal_p_vs_t(562)=2.12717637550086D-01
normal_p_vs_t(563)=2.11960635593791D-01
normal_p_vs_t(564)=2.11203633637496D-01
normal_p_vs_t(565)=2.10446631681201D-01
normal_p_vs_t(566)=2.09689629724906D-01
normal_p_vs_t(567)=2.08680293225521D-01
normal_p_vs_t(568)=2.07670956726136D-01
normal_p_vs_t(569)=2.06661620226751D-01
normal_p_vs_t(570)=2.06156951977059D-01
normal_p_vs_t(571)=2.05652283727366D-01
normal_p_vs_t(572)=2.05147615477674D-01
normal_p_vs_t(573)=2.04642947227981D-01
normal_p_vs_t(574)=2.04138278978289D-01
normal_p_vs_t(575)=2.03633610728596D-01
normal_p_vs_t(576)=2.02876608354057D-01
normal_p_vs_t(577)=2.02119605979519D-01
normal_p_vs_t(578)=2.01362603604980D-01
normal_p_vs_t(579)=2.00605601230441D-01
normal_p_vs_t(580)=1.99848599274146D-01
normal_p_vs_t(581)=1.99091597317851D-01
normal_p_vs_t(582)=1.98334595361556D-01
normal_p_vs_t(583)=1.97577593405261D-01
normal_p_vs_t(584)=1.97199092217992D-01
normal_p_vs_t(585)=1.96820591030722D-01
normal_p_vs_t(586)=1.96442089843453D-01
```

```
normal_p_vs_t(587)=1.96063588656183D-01
normal_p_vs_t(588)=1.95685087468914D-01
normal_p_vs_t(589)=1.95306586281645D-01
normal_p_vs_t(590)=1.94928085094375D-01
normal_p_vs_t(591)=1.94549583907106D-01
normal_p_vs_t(592)=1.93792581532567D-01
normal_p_vs_t(593)=1.93035579158028D-01
normal_p_vs_t(594)=1.92278576783490D-01
normal_p_vs_t(595)=1.91521574408951D-01
normal_p_vs_t(596)=1.90764572034412D-01
normal_p_vs_t(597)=1.90007569659873D-01
normal_p_vs_t(598)=1.89250567285335D-01
normal_p_vs_t(599)=1.88493564910796D-01
normal_p_vs_t(600)=1.87736562954501D-01
normal_p_vs_t(601)=1.86979560998206D-01
normal_p_vs_t(602)=1.86222559041911D-01
normal_p_vs_t(603)=1.85465557085616D-01
normal_p_vs_t(604)=1.85032984300165D-01
normal_p_vs_t(605)=1.84600411514714D-01
normal_p_vs_t(606)=1.84167838729264D-01
normal_p_vs_t(607)=1.83735265943813D-01
normal_p_vs_t(608)=1.83302693158362D-01
normal_p_vs_t(609)=1.82870120372911D-01
normal_p_vs_t(610)=1.82437547587461D-01
normal_p_vs_t(611)=1.81831945687830D-01
normal_p_vs_t(612)=1.81226343788199D-01
normal_p_vs_t(613)=1.80620741888568D-01
normal_p_vs_t(614)=1.80015139988937D-01
normal_p_vs_t(615)=1.79409538089306D-01
normal_p_vs_t(616)=1.78803936189675D-01
normal_p_vs_t(617)=1.78198334290044D-01
normal_p_vs_t(618)=1.77592732390413D-01
normal_p_vs_t(619)=1.76987130490782D-01
normal_p_vs_t(620)=1.76381528591151D-01
normal_p_vs_t(621)=1.75624526634856D-01
normal_p_vs_t(622)=1.74867524678561D-01
normal_p_vs_t(623)=1.74110522722266D-01
normal_p_vs_t(624)=1.73353520765971D-01
normal_p_vs_t(625)=1.73137234373245D-01
normal_p_vs_t(626)=1.72920947980520D-01
normal_p_vs_t(627)=1.72704661587794D-01
normal_p_vs_t(628)=1.72488375195069D-01
normal_p_vs_t(629)=1.72272088802344D-01
normal_p_vs_t(630)=1.72055802409618D-01
normal_p_vs_t(631)=1.71322761395858D-01
normal_p_vs_t(632)=1.70589720382097D-01
!----------------------------------------------
```

```
!-----------------------------------------------
normal_position_p_vs_x(1)=0.00000000000000D+00
normal_position_p_vs_x(2)=7.61164635120030D-04
normal_position_p_vs_x(3)=2.28349390536009D-03
normal_position_p_vs_x(4)=3.80582317560015D-03
normal_position_p_vs_x(5)=5.33347527545643D-03
normal_position_p_vs_x(6)=6.85580454569649D-03
normal_position_p_vs_x(7)=8.37813381593655D-03
normal_position_p_vs_x(8)=9.90046308617661D-03
normal_position_p_vs_x(9)=1.14227923564167D-02
normal_position_p_vs_x(10)=1.29451216266567D-02
normal_position_p_vs_x(11)=1.44727737265130D-02
normal_position_p_vs_x(12)=1.59951029967531D-02
normal_position_p_vs_x(13)=1.75174322669931D-02
normal_position_p_vs_x(14)=1.90397615372332D-02
normal_position_p_vs_x(15)=2.05567679778570D-02
normal_position_p_vs_x(16)=2.20844200777133D-02
normal_position_p_vs_x(17)=2.36120721775696D-02
normal_position_p_vs_x(18)=2.51344014478097D-02
normal_position_p_vs_x(19)=2.66567307180497D-02
normal_position_p_vs_x(20)=2.81737371586736D-02
normal_position_p_vs_x(21)=2.97013892585298D-02
normal_position_p_vs_x(22)=3.12237185287699D-02
normal_position_p_vs_x(23)=3.27513706286262D-02
normal_position_p_vs_x(24)=3.42736998988662D-02
normal_position_p_vs_x(25)=3.57907063394901D-02
normal_position_p_vs_x(26)=3.73130356097301D-02
normal_position_p_vs_x(27)=3.88406877095864D-02
normal_position_p_vs_x(28)=4.03630169798265D-02
normal_position_p_vs_x(29)=4.18906690796828D-02
normal_position_p_vs_x(30)=4.34076755203066D-02
normal_position_p_vs_x(31)=4.49300047905467D-02
normal_position_p_vs_x(32)=4.64576568904029D-02
normal_position_p_vs_x(33)=4.79799861606430D-02
normal_position_p_vs_x(34)=4.95023154308831D-02
normal_position_p_vs_x(35)=5.10246447011231D-02
normal_position_p_vs_x(36)=5.25469739713632D-02
normal_position_p_vs_x(37)=5.40746260712195D-02
normal_position_p_vs_x(38)=5.55969553414595D-02
normal_position_p_vs_x(39)=5.71192846116996D-02
normal_position_p_vs_x(40)=5.86416138819396D-02
normal_position_p_vs_x(41)=6.01692659817959D-02
normal_position_p_vs_x(42)=6.16915952520360D-02
normal_position_p_vs_x(43)=6.32139245222760D-02
normal_position_p_vs_x(44)=6.47362537925161D-02
normal_position_p_vs_x(45)=6.62585830627562D-02
```

```
normal_position_p_vs_x(46)=6.77809123329962D-02
normal_position_p_vs_x(47)=6.93085644328525D-02
normal_position_p_vs_x(48)=7.08308937030926D-02
normal_position_p_vs_x(49)=7.23532229733326D-02
normal_position_p_vs_x(50)=7.38755522435727D-02
normal_position_p_vs_x(51)=7.53978815138128D-02
normal_position_p_vs_x(52)=7.69255336136690D-02
normal_position_p_vs_x(53)=7.84478628839091D-02
normal_position_p_vs_x(54)=7.99701921541491D-02
normal_position_p_vs_x(55)=8.14925214243892D-02
normal_position_p_vs_x(56)=8.30148506946293D-02
normal_position_p_vs_x(57)=8.45371799648693D-02
normal_position_p_vs_x(58)=8.60648320647256D-02
normal_position_p_vs_x(59)=8.75871613349657D-02
normal_position_p_vs_x(60)=8.91094906052057D-02
normal_position_p_vs_x(61)=9.06318198754458D-02
normal_position_p_vs_x(62)=9.21541491456859D-02
normal_position_p_vs_x(63)=9.36764784159259D-02
normal_position_p_vs_x(64)=9.52041305157822D-02
normal_position_p_vs_x(65)=9.67264597860222D-02
normal_position_p_vs_x(66)=9.82487890562623D-02
normal_position_p_vs_x(67)=9.97711183265024D-02
normal_position_p_vs_x(68)=1.01293447596742D-01
normal_position_p_vs_x(69)=1.02815776866982D-01
normal_position_p_vs_x(70)=1.04343428966839D-01
normal_position_p_vs_x(71)=1.05865758237079D-01
normal_position_p_vs_x(72)=1.07388087507319D-01
normal_position_p_vs_x(73)=1.08910416777559D-01
normal_position_p_vs_x(74)=1.10432746047799D-01
normal_position_p_vs_x(75)=1.11955075318039D-01
normal_position_p_vs_x(76)=1.13482727417895D-01
normal_position_p_vs_x(77)=1.15005056688135D-01
normal_position_p_vs_x(78)=1.16527385958375D-01
normal_position_p_vs_x(79)=1.18049715228616D-01
normal_position_p_vs_x(80)=1.19572044498856D-01
normal_position_p_vs_x(81)=1.21094373769096D-01
normal_position_p_vs_x(82)=1.22622025868952D-01
normal_position_p_vs_x(83)=1.24144355139192D-01
normal_position_p_vs_x(84)=1.25666684409432D-01
normal_position_p_vs_x(85)=1.27189013679672D-01
normal_position_p_vs_x(86)=1.28711342949912D-01
normal_position_p_vs_x(87)=1.30228349390536D-01
normal_position_p_vs_x(88)=1.31761324320009D-01
normal_position_p_vs_x(89)=1.33283653590249D-01
normal_position_p_vs_x(90)=1.34805982860489D-01
normal_position_p_vs_x(91)=1.36328312130729D-01
normal_position_p_vs_x(92)=1.37845318571353D-01
normal_position_p_vs_x(93)=1.39367647841593D-01
```

```
normal_position_p_vs_x(94)=1.40900622771065D-01
normal_position_p_vs_x(95)=1.42422952041305D-01
normal_position_p_vs_x(96)=1.43945281311545D-01
normal_position_p_vs_x(97)=1.45462287752169D-01
normal_position_p_vs_x(98)=1.46984617022409D-01
normal_position_p_vs_x(99)=1.48512269122265D-01
normal_position_p_vs_x(100)=1.50039921222122D-01
normal_position_p_vs_x(101)=1.51562250492362D-01
normal_position_p_vs_x(102)=1.53084579762602D-01
normal_position_p_vs_x(103)=1.54601586203226D-01
normal_position_p_vs_x(104)=1.56129238303082D-01
normal_position_p_vs_x(105)=1.57651567573322D-01
normal_position_p_vs_x(106)=1.59179219673178D-01
normal_position_p_vs_x(107)=1.60701548943418D-01
normal_position_p_vs_x(108)=1.62223878213658D-01
normal_position_p_vs_x(109)=1.63746207483898D-01
normal_position_p_vs_x(110)=1.65268536754139D-01
normal_position_p_vs_x(111)=1.66790866024379D-01
normal_position_p_vs_x(112)=1.68318518124235D-01
normal_position_p_vs_x(113)=1.69840847394475D-01
normal_position_p_vs_x(114)=1.71363176664715D-01
normal_position_p_vs_x(115)=1.72885505934955D-01
normal_position_p_vs_x(116)=1.74407835205195D-01
normal_position_p_vs_x(117)=1.75930164475435D-01
normal_position_p_vs_x(118)=1.77452493745675D-01
normal_position_p_vs_x(119)=1.78980145845531D-01
normal_position_p_vs_x(120)=1.80502475115772D-01
normal_position_p_vs_x(121)=1.82024804386012D-01
normal_position_p_vs_x(122)=1.83547133656252D-01
normal_position_p_vs_x(123)=1.85069462926492D-01
normal_position_p_vs_x(124)=1.86597115026348D-01
normal_position_p_vs_x(125)=1.88119444296588D-01
normal_position_p_vs_x(126)=1.89641773566828D-01
normal_position_p_vs_x(127)=1.91164102837068D-01
normal_position_p_vs_x(128)=1.92686432107308D-01
normal_position_p_vs_x(129)=1.94208761377548D-01
normal_position_p_vs_x(130)=1.95736413477405D-01
normal_position_p_vs_x(131)=1.97258742747645D-01
normal_position_p_vs_x(132)=1.98781072017885D-01
normal_position_p_vs_x(133)=2.00303401288125D-01
normal_position_p_vs_x(134)=2.01825730558365D-01
normal_position_p_vs_x(135)=2.03348059828605D-01
normal_position_p_vs_x(136)=2.04875711928461D-01
normal_position_p_vs_x(137)=2.06398041198701D-01
normal_position_p_vs_x(138)=2.07920370468941D-01
normal_position_p_vs_x(139)=2.09442699739181D-01
normal_position_p_vs_x(140)=2.10965029009421D-01
normal_position_p_vs_x(141)=2.12492681109278D-01
```

```
normal_position_p_vs_x(142)=2.14015010379518D-01
normal_position_p_vs_x(143)=2.15537339649758D-01
normal_position_p_vs_x(144)=2.17059668919998D-01
normal_position_p_vs_x(145)=2.18581998190238D-01
normal_position_p_vs_x(146)=2.20104327460478D-01
normal_position_p_vs_x(147)=2.21631979560334D-01
normal_position_p_vs_x(148)=2.23154308830574D-01
normal_position_p_vs_x(149)=2.24676638100814D-01
normal_position_p_vs_x(150)=2.26198967371054D-01
normal_position_p_vs_x(151)=2.27721296641295D-01
normal_position_p_vs_x(152)=2.29243625911535D-01
normal_position_p_vs_x(153)=2.30771278011391D-01
normal_position_p_vs_x(154)=2.32293607281631D-01
normal_position_p_vs_x(155)=2.33815936551871D-01
normal_position_p_vs_x(156)=2.35338265822111D-01
normal_position_p_vs_x(157)=2.36860595092351D-01
normal_position_p_vs_x(158)=2.38382924362591D-01
normal_position_p_vs_x(159)=2.39910576462447D-01
normal_position_p_vs_x(160)=2.41432905732688D-01
normal_position_p_vs_x(161)=2.42955235002928D-01
normal_position_p_vs_x(162)=2.44477564273168D-01
normal_position_p_vs_x(163)=2.45999893543408D-01
normal_position_p_vs_x(164)=2.47522222813648D-01
normal_position_p_vs_x(165)=2.49049874913504D-01
normal_position_p_vs_x(166)=2.50572204183744D-01
normal_position_p_vs_x(167)=2.52094533453984D-01
normal_position_p_vs_x(168)=2.53616862724224D-01
normal_position_p_vs_x(169)=2.55139191994464D-01
normal_position_p_vs_x(170)=2.56656198435088D-01
normal_position_p_vs_x(171)=2.58189173364561D-01
normal_position_p_vs_x(172)=2.59711502634801D-01
normal_position_p_vs_x(173)=2.61233831905041D-01
normal_position_p_vs_x(174)=2.62756161175281D-01
normal_position_p_vs_x(175)=2.64273167615905D-01
normal_position_p_vs_x(176)=2.65800819715761D-01
normal_position_p_vs_x(177)=2.67328471815617D-01
normal_position_p_vs_x(178)=2.68850801085857D-01
normal_position_p_vs_x(179)=2.70373130356097D-01
normal_position_p_vs_x(180)=2.71895459626337D-01
normal_position_p_vs_x(181)=2.73417788896577D-01
normal_position_p_vs_x(182)=2.74940118166818D-01
normal_position_p_vs_x(183)=2.76467770266674D-01
normal_position_p_vs_x(184)=2.77990099536914D-01
normal_position_p_vs_x(185)=2.79507105977538D-01
normal_position_p_vs_x(186)=2.81034758077394D-01
normal_position_p_vs_x(187)=2.82557087347634D-01
normal_position_p_vs_x(188)=2.84079416617874D-01
normal_position_p_vs_x(189)=2.85607068717730D-01
```

```
normal_position_p_vs_x(190)=2.87124075158354D-01
normal_position_p_vs_x(191)=2.88651727258210D-01
normal_position_p_vs_x(192)=2.90174056528451D-01
normal_position_p_vs_x(193)=2.91696385798691D-01
normal_position_p_vs_x(194)=2.93218715068931D-01
normal_position_p_vs_x(195)=2.94741044339171D-01
normal_position_p_vs_x(196)=2.96263373609411D-01
normal_position_p_vs_x(197)=2.97791025709267D-01
normal_position_p_vs_x(198)=2.99313354979507D-01
normal_position_p_vs_x(199)=3.00835684249747D-01
normal_position_p_vs_x(200)=3.02358013519987D-01
normal_position_p_vs_x(201)=3.03880342790227D-01
normal_position_p_vs_x(202)=3.05407994890084D-01
normal_position_p_vs_x(203)=3.06930324160324D-01
normal_position_p_vs_x(204)=3.08452653430564D-01
normal_position_p_vs_x(205)=3.09974982700804D-01
normal_position_p_vs_x(206)=3.11497311971044D-01
normal_position_p_vs_x(207)=3.13024964070900D-01
normal_position_p_vs_x(208)=3.14547293341140D-01
normal_position_p_vs_x(209)=3.16069622611380D-01
normal_position_p_vs_x(210)=3.17591951881620D-01
normal_position_p_vs_x(211)=3.19114281151860D-01
normal_position_p_vs_x(212)=3.20636610422100D-01
normal_position_p_vs_x(213)=3.22164262521957D-01
normal_position_p_vs_x(214)=3.23686591792197D-01
normal_position_p_vs_x(215)=3.25208921062437D-01
normal_position_p_vs_x(216)=3.26731250332677D-01
normal_position_p_vs_x(217)=3.28253579602917D-01
normal_position_p_vs_x(218)=3.29775908873157D-01
normal_position_p_vs_x(219)=3.31303560973013D-01
normal_position_p_vs_x(220)=3.32825890243253D-01
normal_position_p_vs_x(221)=3.34348219513493D-01
normal_position_p_vs_x(222)=3.35870548783733D-01
normal_position_p_vs_x(223)=3.37392878053973D-01
normal_position_p_vs_x(224)=3.38915207324214D-01
normal_position_p_vs_x(225)=3.40442859424070D-01
normal_position_p_vs_x(226)=3.41965188694310D-01
normal_position_p_vs_x(227)=3.43487517964550D-01
normal_position_p_vs_x(228)=3.45009847234790D-01
normal_position_p_vs_x(229)=3.46532176505030D-01
normal_position_p_vs_x(230)=3.48059828604886D-01
normal_position_p_vs_x(231)=3.49582157875126D-01
normal_position_p_vs_x(232)=3.51104487145366D-01
normal_position_p_vs_x(233)=3.52626816415607D-01
normal_position_p_vs_x(234)=3.54149145685847D-01
normal_position_p_vs_x(235)=3.55671474956087D-01
normal_position_p_vs_x(236)=3.57199127055943D-01
normal_position_p_vs_x(237)=3.58721456326183D-01
```

```
normal_position_p_vs_x(238)=3.60243785596423D-01
normal_position_p_vs_x(239)=3.61766114866663D-01
normal_position_p_vs_x(240)=3.63288444136903D-01
normal_position_p_vs_x(241)=3.64810773407143D-01
normal_position_p_vs_x(242)=3.66338425507000D-01
normal_position_p_vs_x(243)=3.67860754777240D-01
normal_position_p_vs_x(244)=3.69383084047480D-01
normal_position_p_vs_x(245)=3.70905413317720D-01
normal_position_p_vs_x(246)=3.72427742587960D-01
normal_position_p_vs_x(247)=3.73950071858200D-01
normal_position_p_vs_x(248)=3.75477723958056D-01
normal_position_p_vs_x(249)=3.77000053228296D-01
normal_position_p_vs_x(250)=3.78522382498536D-01
normal_position_p_vs_x(251)=3.80044711768776D-01
normal_position_p_vs_x(252)=3.81567041039016D-01
normal_position_p_vs_x(253)=3.83089370309256D-01
normal_position_p_vs_x(254)=3.84617022409113D-01
normal_position_p_vs_x(255)=3.86139351679353D-01
normal_position_p_vs_x(256)=3.87661680949593D-01
normal_position_p_vs_x(257)=3.89178687390217D-01
normal_position_p_vs_x(258)=3.90706339490073D-01
normal_position_p_vs_x(259)=3.92228668760313D-01
normal_position_p_vs_x(260)=3.93756320860169D-01
normal_position_p_vs_x(261)=3.95278650130409D-01
normal_position_p_vs_x(262)=3.96795656571033D-01
normal_position_p_vs_x(263)=3.98323308670889D-01
normal_position_p_vs_x(264)=3.99845637941130D-01
normal_position_p_vs_x(265)=4.01367967211370D-01
normal_position_p_vs_x(266)=4.02895619311226D-01
normal_position_p_vs_x(267)=4.04412625751850D-01
normal_position_p_vs_x(268)=4.05934955022090D-01
normal_position_p_vs_x(269)=4.07462607121946D-01
normal_position_p_vs_x(270)=4.08984936392186D-01
normal_position_p_vs_x(271)=4.10507265662426D-01
normal_position_p_vs_x(272)=4.12034917762282D-01
normal_position_p_vs_x(273)=4.13551924202906D-01
normal_position_p_vs_x(274)=4.15079576302763D-01
normal_position_p_vs_x(275)=4.16601905573003D-01
normal_position_p_vs_x(276)=4.18124234843243D-01
normal_position_p_vs_x(277)=4.19646564113483D-01
normal_position_p_vs_x(278)=4.21168893383723D-01
normal_position_p_vs_x(279)=4.22696545483579D-01
normal_position_p_vs_x(280)=4.24218874753819D-01
normal_position_p_vs_x(281)=4.25741204024059D-01
normal_position_p_vs_x(282)=4.27263533294299D-01
normal_position_p_vs_x(283)=4.28785862564539D-01
normal_position_p_vs_x(284)=4.30313514664396D-01
normal_position_p_vs_x(285)=4.31835843934636D-01
```

```
normal_position_p_vs_x(286)=4.33358173204876D-01
normal_position_p_vs_x(287)=4.34880502475116D-01
normal_position_p_vs_x(288)=4.36402831745356D-01
normal_position_p_vs_x(289)=4.37925161015596D-01
normal_position_p_vs_x(290)=4.39452813115452D-01
normal_position_p_vs_x(291)=4.40975142385692D-01
normal_position_p_vs_x(292)=4.42497471655932D-01
normal_position_p_vs_x(293)=4.44019800926172D-01
normal_position_p_vs_x(294)=4.45542130196412D-01
normal_position_p_vs_x(295)=4.47064459466652D-01
normal_position_p_vs_x(296)=4.48592111566509D-01
normal_position_p_vs_x(297)=4.50114440836749D-01
normal_position_p_vs_x(298)=4.51636770106989D-01
normal_position_p_vs_x(299)=4.53159099377229D-01
normal_position_p_vs_x(300)=4.54681428647469D-01
normal_position_p_vs_x(301)=4.56203757917709D-01
normal_position_p_vs_x(302)=4.57731410017565D-01
normal_position_p_vs_x(303)=4.59253739287805D-01
normal_position_p_vs_x(304)=4.60776068558045D-01
normal_position_p_vs_x(305)=4.62298397828286D-01
normal_position_p_vs_x(306)=4.63820727098526D-01
normal_position_p_vs_x(307)=4.65343056368766D-01
normal_position_p_vs_x(308)=4.66870708468622D-01
normal_position_p_vs_x(309)=4.68393037738862D-01
normal_position_p_vs_x(310)=4.69915367009102D-01
normal_position_p_vs_x(311)=4.71437696279342D-01
normal_position_p_vs_x(312)=4.72960025549582D-01
normal_position_p_vs_x(313)=4.74482354819822D-01
normal_position_p_vs_x(314)=4.76010006919679D-01
normal_position_p_vs_x(315)=4.77532336189919D-01
normal_position_p_vs_x(316)=4.79054665460159D-01
normal_position_p_vs_x(317)=4.80576994730399D-01
normal_position_p_vs_x(318)=4.82099324000639D-01
normal_position_p_vs_x(319)=4.83626976100495D-01
normal_position_p_vs_x(320)=4.85149305370735D-01
normal_position_p_vs_x(321)=4.86671634640975D-01
normal_position_p_vs_x(322)=4.88193963911215D-01
normal_position_p_vs_x(323)=4.89716293181455D-01
normal_position_p_vs_x(324)=4.91233299622079D-01
normal_position_p_vs_x(325)=4.92766274551552D-01
normal_position_p_vs_x(326)=4.94288603821792D-01
normal_position_p_vs_x(327)=4.95810933092032D-01
normal_position_p_vs_x(328)=4.97333262362272D-01
normal_position_p_vs_x(329)=4.98850268802896D-01
normal_position_p_vs_x(330)=5.00377920902752D-01
normal_position_p_vs_x(331)=5.01905573002608D-01
normal_position_p_vs_x(332)=5.03427902272848D-01
normal_position_p_vs_x(333)=5.04950231543088D-01
```

```
normal_position_p_vs_x(334)=5.06472560813328D-01
normal_position_p_vs_x(335)=5.07989567253952D-01
normal_position_p_vs_x(336)=5.09517219353809D-01
normal_position_p_vs_x(337)=5.11044871453665D-01
normal_position_p_vs_x(338)=5.12567200723905D-01
normal_position_p_vs_x(339)=5.14089529994145D-01
normal_position_p_vs_x(340)=5.15606536434769D-01
normal_position_p_vs_x(341)=5.17134188534625D-01
normal_position_p_vs_x(342)=5.18656517804865D-01
normal_position_p_vs_x(343)=5.20184169904721D-01
normal_position_p_vs_x(344)=5.21706499174961D-01
normal_position_p_vs_x(345)=5.23223505615585D-01
normal_position_p_vs_x(346)=5.24751157715442D-01
normal_position_p_vs_x(347)=5.26273486985682D-01
normal_position_p_vs_x(348)=5.27795816255922D-01
normal_position_p_vs_x(349)=5.29323468355778D-01
normal_position_p_vs_x(350)=5.30840474796402D-01
normal_position_p_vs_x(351)=5.32368126896258D-01
normal_position_p_vs_x(352)=5.33890456166498D-01
normal_position_p_vs_x(353)=5.35412785436738D-01
normal_position_p_vs_x(354)=5.36935114706978D-01
normal_position_p_vs_x(355)=5.38462766806834D-01
normal_position_p_vs_x(356)=5.39985096077075D-01
normal_position_p_vs_x(357)=5.41507425347315D-01
normal_position_p_vs_x(358)=5.43029754617555D-01
normal_position_p_vs_x(359)=5.44552083887795D-01
normal_position_p_vs_x(360)=5.46074413158035D-01
normal_position_p_vs_x(361)=5.47602065257891D-01
normal_position_p_vs_x(362)=5.49124394528131D-01
normal_position_p_vs_x(363)=5.50646723798371D-01
normal_position_p_vs_x(364)=5.52169053068611D-01
normal_position_p_vs_x(365)=5.53691382338851D-01
normal_position_p_vs_x(366)=5.55213711609091D-01
normal_position_p_vs_x(367)=5.56741363708948D-01
normal_position_p_vs_x(368)=5.58263692979188D-01
normal_position_p_vs_x(369)=5.59786022249428D-01
normal_position_p_vs_x(370)=5.61308351519668D-01
normal_position_p_vs_x(371)=5.62830680789908D-01
normal_position_p_vs_x(372)=5.64353010060148D-01
normal_position_p_vs_x(373)=5.65880662160004D-01
normal_position_p_vs_x(374)=5.67402991430244D-01
normal_position_p_vs_x(375)=5.68925320700484D-01
normal_position_p_vs_x(376)=5.70447649970725D-01
normal_position_p_vs_x(377)=5.71969979240964D-01
normal_position_p_vs_x(378)=5.73492308511205D-01
normal_position_p_vs_x(379)=5.75019960611061D-01
normal_position_p_vs_x(380)=5.76542289881301D-01
normal_position_p_vs_x(381)=5.78064619151541D-01
```

```
normal_position_p_vs_x(382)=5.79586948421781D-01
normal_position_p_vs_x(383)=5.81109277692021D-01
normal_position_p_vs_x(384)=5.82631606962261D-01
normal_position_p_vs_x(385)=5.84159259062118D-01
normal_position_p_vs_x(386)=5.85681588332358D-01
normal_position_p_vs_x(387)=5.87203917602598D-01
normal_position_p_vs_x(388)=5.88726246872838D-01
normal_position_p_vs_x(389)=5.90248576143078D-01
normal_position_p_vs_x(390)=5.91770905413318D-01
normal_position_p_vs_x(391)=5.93298557513174D-01
normal_position_p_vs_x(392)=5.94820886783414D-01
normal_position_p_vs_x(393)=5.96343216053654D-01
normal_position_p_vs_x(394)=5.97865545323894D-01
normal_position_p_vs_x(395)=5.99387874594134D-01
normal_position_p_vs_x(396)=6.00910203864374D-01
normal_position_p_vs_x(397)=6.02437855964231D-01
normal_position_p_vs_x(398)=6.03960185234471D-01
normal_position_p_vs_x(399)=6.05482514504711D-01
normal_position_p_vs_x(400)=6.07004843774951D-01
normal_position_p_vs_x(401)=6.08527173045191D-01
normal_position_p_vs_x(402)=6.10049502315431D-01
normal_position_p_vs_x(403)=6.11577154415287D-01
normal_position_p_vs_x(404)=6.13099483685527D-01
normal_position_p_vs_x(405)=6.14621812955767D-01
normal_position_p_vs_x(406)=6.16144142226007D-01
normal_position_p_vs_x(407)=6.17661148666631D-01
normal_position_p_vs_x(408)=6.19194123596104D-01
normal_position_p_vs_x(409)=6.20716452866344D-01
normal_position_p_vs_x(410)=6.22238782136584D-01
normal_position_p_vs_x(411)=6.23761111406824D-01
normal_position_p_vs_x(412)=6.25278117847448D-01
normal_position_p_vs_x(413)=6.26805769947304D-01
normal_position_p_vs_x(414)=6.28333422047160D-01
normal_position_p_vs_x(415)=6.29855751317400D-01
normal_position_p_vs_x(416)=6.31378080587640D-01
normal_position_p_vs_x(417)=6.32895087028264D-01
normal_position_p_vs_x(418)=6.34422739128121D-01
normal_position_p_vs_x(419)=6.35945068398361D-01
normal_position_p_vs_x(420)=6.37472720498217D-01
normal_position_p_vs_x(421)=6.38995049768457D-01
normal_position_p_vs_x(422)=6.40517379038697D-01
normal_position_p_vs_x(423)=6.42039708308937D-01
normal_position_p_vs_x(424)=6.43562037579177D-01
normal_position_p_vs_x(425)=6.45084366849417D-01
normal_position_p_vs_x(426)=6.46612018949273D-01
normal_position_p_vs_x(427)=6.48129025389897D-01
normal_position_p_vs_x(428)=6.49656677489754D-01
normal_position_p_vs_x(429)=6.51179006759994D-01
```

```
normal_position_p_vs_x(430)=6.52701336030234D-01
normal_position_p_vs_x(431)=6.54223665300474D-01
normal_position_p_vs_x(432)=6.55745994570714D-01
normal_position_p_vs_x(433)=6.57273646670570D-01
normal_position_p_vs_x(434)=6.58795975940810D-01
normal_position_p_vs_x(435)=6.60318305211050D-01
normal_position_p_vs_x(436)=6.61840634481290D-01
normal_position_p_vs_x(437)=6.63362963751530D-01
normal_position_p_vs_x(438)=6.64890615851387D-01
normal_position_p_vs_x(439)=6.66412945121627D-01
normal_position_p_vs_x(440)=6.67935274391867D-01
normal_position_p_vs_x(441)=6.69457603662107D-01
normal_position_p_vs_x(442)=6.70979932932347D-01
normal_position_p_vs_x(443)=6.72502262202587D-01
normal_position_p_vs_x(444)=6.74029914302443D-01
normal_position_p_vs_x(445)=6.75552243572683D-01
normal_position_p_vs_x(446)=6.77074572842923D-01
normal_position_p_vs_x(447)=6.78596902113163D-01
normal_position_p_vs_x(448)=6.80119231383403D-01
normal_position_p_vs_x(449)=6.81641560653643D-01
normal_position_p_vs_x(450)=6.83169212753500D-01
normal_position_p_vs_x(451)=6.84691542023740D-01
normal_position_p_vs_x(452)=6.86213871293980D-01
normal_position_p_vs_x(453)=6.87736200564220D-01
normal_position_p_vs_x(454)=6.89258529834460D-01
normal_position_p_vs_x(455)=6.90780859104700D-01
normal_position_p_vs_x(456)=6.92308511204556D-01
normal_position_p_vs_x(457)=6.93830840474796D-01
normal_position_p_vs_x(458)=6.95353169745036D-01
normal_position_p_vs_x(459)=6.96875499015277D-01
normal_position_p_vs_x(460)=6.98397828285517D-01
normal_position_p_vs_x(461)=6.99920157555757D-01
normal_position_p_vs_x(462)=7.01447809655613D-01
normal_position_p_vs_x(463)=7.02970138925853D-01
normal_position_p_vs_x(464)=7.04492468196093D-01
normal_position_p_vs_x(465)=7.06014797466333D-01
normal_position_p_vs_x(466)=7.07537126736573D-01
normal_position_p_vs_x(467)=7.09059456006813D-01
normal_position_p_vs_x(468)=7.10587108106670D-01
normal_position_p_vs_x(469)=7.12109437376910D-01
normal_position_p_vs_x(470)=7.13631766647150D-01
normal_position_p_vs_x(471)=7.15154095917390D-01
normal_position_p_vs_x(472)=7.16676425187630D-01
normal_position_p_vs_x(473)=7.18198754457870D-01
normal_position_p_vs_x(474)=7.19726406557726D-01
normal_position_p_vs_x(475)=7.21248735827966D-01
normal_position_p_vs_x(476)=7.22771065098206D-01
normal_position_p_vs_x(477)=7.24293394368446D-01
```

```
normal_position_p_vs_x(478)=7.25815723638686D-01
normal_position_p_vs_x(479)=7.27332730079310D-01
normal_position_p_vs_x(480)=7.28865705008783D-01
normal_position_p_vs_x(481)=7.30388034279023D-01
normal_position_p_vs_x(482)=7.31910363549263D-01
normal_position_p_vs_x(483)=7.33432692819503D-01
normal_position_p_vs_x(484)=7.34949699260127D-01
normal_position_p_vs_x(485)=7.36477351359983D-01
normal_position_p_vs_x(486)=7.38005003459839D-01
normal_position_p_vs_x(487)=7.39527332730079D-01
normal_position_p_vs_x(488)=7.41049662000319D-01
normal_position_p_vs_x(489)=7.42566668440943D-01
normal_position_p_vs_x(490)=7.44088997711183D-01
normal_position_p_vs_x(491)=7.45616649811040D-01
normal_position_p_vs_x(492)=7.47144301910896D-01
normal_position_p_vs_x(493)=7.48666631181136D-01
normal_position_p_vs_x(494)=7.50183637621760D-01
normal_position_p_vs_x(495)=7.51711289721616D-01
normal_position_p_vs_x(496)=7.53233618991856D-01
normal_position_p_vs_x(497)=7.54761271091712D-01
normal_position_p_vs_x(498)=7.56283600361952D-01
normal_position_p_vs_x(499)=7.57800606802576D-01
normal_position_p_vs_x(500)=7.59328258902432D-01
normal_position_p_vs_x(501)=7.60850588172673D-01
normal_position_p_vs_x(502)=7.62372917442913D-01
normal_position_p_vs_x(503)=7.63900569542769D-01
normal_position_p_vs_x(504)=7.65422898813009D-01
normal_position_p_vs_x(505)=7.66945228083249D-01
normal_position_p_vs_x(506)=7.68467557353489D-01
normal_position_p_vs_x(507)=7.69989886623729D-01
normal_position_p_vs_x(508)=7.71512215893969D-01
normal_position_p_vs_x(509)=7.73039867993826D-01
normal_position_p_vs_x(510)=7.74556874434449D-01
normal_position_p_vs_x(511)=7.76084526534306D-01
normal_position_p_vs_x(512)=7.77606855804546D-01
normal_position_p_vs_x(513)=7.79129185074786D-01
normal_position_p_vs_x(514)=7.80651514345026D-01
normal_position_p_vs_x(515)=7.82173843615266D-01
normal_position_p_vs_x(516)=7.83701495715122D-01
normal_position_p_vs_x(517)=7.85223824985362D-01
normal_position_p_vs_x(518)=7.86746154255602D-01
normal_position_p_vs_x(519)=7.88268483525842D-01
normal_position_p_vs_x(520)=7.89790812796082D-01
normal_position_p_vs_x(521)=7.91318464895939D-01
normal_position_p_vs_x(522)=7.92840794166179D-01
normal_position_p_vs_x(523)=7.94363123436419D-01
normal_position_p_vs_x(524)=7.95885452706659D-01
normal_position_p_vs_x(525)=7.97407781976899D-01
```

```
normal_position_p_vs_x(526)=7.98930111247139D-01
normal_position_p_vs_x(527)=8.00457763346995D-01
normal_position_p_vs_x(528)=8.01980092617235D-01
normal_position_p_vs_x(529)=8.03502421887475D-01
normal_position_p_vs_x(530)=8.05024751157716D-01
normal_position_p_vs_x(531)=8.06547080427956D-01
normal_position_p_vs_x(532)=8.08069409698196D-01
normal_position_p_vs_x(533)=8.09597061798052D-01
normal_position_p_vs_x(534)=8.11119391068292D-01
normal_position_p_vs_x(535)=8.12641720338532D-01
normal_position_p_vs_x(536)=8.14164049608772D-01
normal_position_p_vs_x(537)=8.15686378879012D-01
normal_position_p_vs_x(538)=8.17208708149252D-01
normal_position_p_vs_x(539)=8.18736360249108D-01
normal_position_p_vs_x(540)=8.20258689519349D-01
normal_position_p_vs_x(541)=8.21781018789589D-01
normal_position_p_vs_x(542)=8.23303348059829D-01
normal_position_p_vs_x(543)=8.24825677330069D-01
normal_position_p_vs_x(544)=8.26348006600309D-01
normal_position_p_vs_x(545)=8.27875658700165D-01
normal_position_p_vs_x(546)=8.29397987970405D-01
normal_position_p_vs_x(547)=8.30920317240645D-01
normal_position_p_vs_x(548)=8.32442646510885D-01
normal_position_p_vs_x(549)=8.33964975781125D-01
normal_position_p_vs_x(550)=8.35487305051365D-01
normal_position_p_vs_x(551)=8.37014957151222D-01
normal_position_p_vs_x(552)=8.38537286421462D-01
normal_position_p_vs_x(553)=8.40059615691702D-01
normal_position_p_vs_x(554)=8.41581944961942D-01
normal_position_p_vs_x(555)=8.43104274232182D-01
normal_position_p_vs_x(556)=8.44621280672806D-01
normal_position_p_vs_x(557)=8.46154255602278D-01
normal_position_p_vs_x(558)=8.47676584872518D-01
normal_position_p_vs_x(559)=8.49198914142758D-01
normal_position_p_vs_x(560)=8.50721243412998D-01
normal_position_p_vs_x(561)=8.52243572683238D-01
normal_position_p_vs_x(562)=8.53765901953478D-01
normal_position_p_vs_x(563)=8.55293554053335D-01
normal_position_p_vs_x(564)=8.56815883323575D-01
normal_position_p_vs_x(565)=8.58338212593815D-01
normal_position_p_vs_x(566)=8.59860541864055D-01
normal_position_p_vs_x(567)=8.61382871134295D-01
normal_position_p_vs_x(568)=8.62905200404535D-01
normal_position_p_vs_x(569)=8.64432852504392D-01
normal_position_p_vs_x(570)=8.65955181774631D-01
normal_position_p_vs_x(571)=8.67477511044871D-01
normal_position_p_vs_x(572)=8.68999840315112D-01
normal_position_p_vs_x(573)=8.70522169585352D-01
```

```
normal_position_p_vs_x(574)=8.72044498855592D-01
normal_position_p_vs_x(575)=8.73572150955448D-01
normal_position_p_vs_x(576)=8.75094480225688D-01
normal_position_p_vs_x(577)=8.76611486666312D-01
normal_position_p_vs_x(578)=8.78139138766168D-01
normal_position_p_vs_x(579)=8.79661468036408D-01
normal_position_p_vs_x(580)=8.81183797306648D-01
normal_position_p_vs_x(581)=8.82711449406505D-01
normal_position_p_vs_x(582)=8.84228455847128D-01
normal_position_p_vs_x(583)=8.85756107946985D-01
normal_position_p_vs_x(584)=8.87278437217225D-01
normal_position_p_vs_x(585)=8.88800766487465D-01
normal_position_p_vs_x(586)=8.90328418587321D-01
normal_position_p_vs_x(587)=8.91845425027945D-01
normal_position_p_vs_x(588)=8.93373077127801D-01
normal_position_p_vs_x(589)=8.94895406398041D-01
normal_position_p_vs_x(590)=8.96417735668281D-01
normal_position_p_vs_x(591)=8.97940064938521D-01
normal_position_p_vs_x(592)=8.99462394208761D-01
normal_position_p_vs_x(593)=9.00984723479001D-01
normal_position_p_vs_x(594)=9.02512375578858D-01
normal_position_p_vs_x(595)=9.04034704849098D-01
normal_position_p_vs_x(596)=9.05557034119338D-01
normal_position_p_vs_x(597)=9.07079363389578D-01
normal_position_p_vs_x(598)=9.08607015489434D-01
normal_position_p_vs_x(599)=9.10129344759674D-01
normal_position_p_vs_x(600)=9.11651674029914D-01
normal_position_p_vs_x(601)=9.13174003300154D-01
normal_position_p_vs_x(602)=9.14696332570395D-01
normal_position_p_vs_x(603)=9.16218661840635D-01
normal_position_p_vs_x(604)=9.17746313940491D-01
normal_position_p_vs_x(605)=9.19268643210731D-01
normal_position_p_vs_x(606)=9.20790972480971D-01
normal_position_p_vs_x(607)=9.22313301751211D-01
normal_position_p_vs_x(608)=9.23835631021451D-01
normal_position_p_vs_x(609)=9.25357960291691D-01
normal_position_p_vs_x(610)=9.26885612391547D-01
normal_position_p_vs_x(611)=9.28407941661787D-01
normal_position_p_vs_x(612)=9.29930270932028D-01
normal_position_p_vs_x(613)=9.31452600202267D-01
normal_position_p_vs_x(614)=9.32974929472508D-01
normal_position_p_vs_x(615)=9.34497258742748D-01
normal_position_p_vs_x(616)=9.36024910842604D-01
normal_position_p_vs_x(617)=9.37547240112844D-01
normal_position_p_vs_x(618)=9.39069569383084D-01
normal_position_p_vs_x(619)=9.40591898653324D-01
normal_position_p_vs_x(620)=9.42114227923564D-01
normal_position_p_vs_x(621)=9.43636557193804D-01
```

```
normal_position_p_vs_x(622)=9.45164209293661D-01
normal_position_p_vs_x(623)=9.46686538563901D-01
normal_position_p_vs_x(624)=9.48208867834141D-01
normal_position_p_vs_x(625)=9.49731197104381D-01
normal_position_p_vs_x(626)=9.51253526374621D-01
normal_position_p_vs_x(627)=9.52775855644861D-01
normal_position_p_vs_x(628)=9.54303507744717D-01
normal_position_p_vs_x(629)=9.55825837014957D-01
normal_position_p_vs_x(630)=9.57348166285197D-01
normal_position_p_vs_x(631)=9.58870495555437D-01
normal_position_p_vs_x(632)=9.60392824825677D-01
normal_position_p_vs_x(633)=9.61915154095917D-01
normal_position_p_vs_x(634)=9.63442806195774D-01
normal_position_p_vs_x(635)=9.64965135466014D-01
normal_position_p_vs_x(636)=9.66487464736254D-01
normal_position_p_vs_x(637)=9.68009794006494D-01
normal_position_p_vs_x(638)=9.69532123276734D-01
normal_position_p_vs_x(639)=9.71054452546974D-01
normal_position_p_vs_x(640)=9.72582104646830D-01
normal_position_p_vs_x(641)=9.74104433917070D-01
normal_position_p_vs_x(642)=9.75626763187310D-01
normal_position_p_vs_x(643)=9.77149092457551D-01
normal_position_p_vs_x(644)=9.78666098898174D-01
normal_position_p_vs_x(645)=9.80193750998031D-01
normal_position_p_vs_x(646)=9.81721403097887D-01
normal_position_p_vs_x(647)=9.83243732368127D-01
normal_position_p_vs_x(648)=9.84766061638367D-01
normal_position_p_vs_x(649)=9.86283068078991D-01
normal_position_p_vs_x(650)=9.87810720178847D-01
normal_position_p_vs_x(651)=9.89333049449087D-01
normal_position_p_vs_x(652)=9.90860701548944D-01
normal_position_p_vs_x(653)=9.92383030819183D-01
normal_position_p_vs_x(654)=9.93900037259807D-01
normal_position_p_vs_x(655)=9.95422366530047D-01
normal_position_p_vs_x(656)=9.96950018629904D-01
normal_position_p_vs_x(657)=9.98472347900144D-01
normal_position_p_vs_x(658)=1.00000000000000D+00
!------------------------------------------------



!------------------------------------------------
normal_time_p_vs_t(1)=0.00000000000000D+00
normal_time_p_vs_t(2)=1.12114730189597D-03
normal_time_p_vs_t(3)=2.64297520661157D-03
normal_time_p_vs_t(4)=4.16567817209528D-03
normal_time_p_vs_t(5)=5.68847836655323D-03
```

```
normal_time_p_vs_t(6)=7.21030627126884D-03
normal_time_p_vs_t(7)=8.73203694701021D-03
normal_time_p_vs_t(8)=1.02090422946038D-02
normal_time_p_vs_t(9)=1.17647058823529D-02
normal_time_p_vs_t(10)=1.33203694701021D-02
normal_time_p_vs_t(11)=1.47788040836169D-02
normal_time_p_vs_t(12)=1.63344676713661D-02
normal_time_p_vs_t(13)=1.78901312591152D-02
normal_time_p_vs_t(14)=1.93485658726300D-02
normal_time_p_vs_t(15)=2.09042294603792D-02
normal_time_p_vs_t(16)=2.24598930481283D-02
normal_time_p_vs_t(17)=2.39183276616432D-02
normal_time_p_vs_t(18)=2.54739912493923D-02
normal_time_p_vs_t(19)=2.70296548371415D-02
normal_time_p_vs_t(20)=2.84880894506563D-02
normal_time_p_vs_t(21)=3.00437530384054D-02
normal_time_p_vs_t(22)=3.15994166261546D-02
normal_time_p_vs_t(23)=3.30578512396694D-02
normal_time_p_vs_t(24)=3.46135148274186D-02
normal_time_p_vs_t(25)=3.61691784151677D-02
normal_time_p_vs_t(26)=3.76276130286825D-02
normal_time_p_vs_t(27)=3.91832766164317D-02
normal_time_p_vs_t(28)=4.07389402041808D-02
normal_time_p_vs_t(29)=4.21973748176957D-02
normal_time_p_vs_t(30)=4.37530384054448D-02
normal_time_p_vs_t(31)=4.53087019931940D-02
normal_time_p_vs_t(32)=4.67671366067088D-02
normal_time_p_vs_t(33)=4.83228001944579D-02
normal_time_p_vs_t(34)=4.98784637822071D-02
normal_time_p_vs_t(35)=5.13368983957219D-02
normal_time_p_vs_t(36)=5.28925619834711D-02
normal_time_p_vs_t(37)=5.43509965969859D-02
normal_time_p_vs_t(38)=5.59066601847351D-02
normal_time_p_vs_t(39)=5.89207583859990D-02
normal_time_p_vs_t(40)=6.04764219737482D-02
normal_time_p_vs_t(41)=6.20320855614973D-02
normal_time_p_vs_t(42)=6.34905201750122D-02
normal_time_p_vs_t(43)=6.50461837627613D-02
normal_time_p_vs_t(44)=6.66018473505105D-02
normal_time_p_vs_t(45)=6.80602819640253D-02
normal_time_p_vs_t(46)=6.96159455517744D-02
normal_time_p_vs_t(47)=7.26300437530384D-02
normal_time_p_vs_t(48)=7.71998055420515D-02
normal_time_p_vs_t(49)=8.03111327175498D-02
normal_time_p_vs_t(50)=8.17695673310646D-02
normal_time_p_vs_t(51)=8.33252309188138D-02
normal_time_p_vs_t(52)=8.47836655323286D-02
normal_time_p_vs_t(53)=8.63393291200778D-02
```

```
normal_time_p_vs_t(54)=8.78949927078269D-02
normal_time_p_vs_t(55)=8.93534273213418D-02
normal_time_p_vs_t(56)=9.09090909090909D-02
normal_time_p_vs_t(57)=9.24647544968401D-02
normal_time_p_vs_t(58)=9.39231891103549D-02
normal_time_p_vs_t(59)=9.54788526981040D-02
normal_time_p_vs_t(60)=9.70345162858532D-02
normal_time_p_vs_t(61)=9.84929508993680D-02
normal_time_p_vs_t(62)=1.00048614487117D-01
normal_time_p_vs_t(63)=1.01604278074866D-01
normal_time_p_vs_t(64)=1.03062712688381D-01
normal_time_p_vs_t(65)=1.04618376276130D-01
normal_time_p_vs_t(66)=1.06174039863879D-01
normal_time_p_vs_t(67)=1.07632474477394D-01
normal_time_p_vs_t(68)=1.09188138065143D-01
normal_time_p_vs_t(69)=1.10646572678658D-01
normal_time_p_vs_t(70)=1.12202236266407D-01
normal_time_p_vs_t(71)=1.13757899854157D-01
normal_time_p_vs_t(72)=1.15216334467671D-01
normal_time_p_vs_t(73)=1.16771998055421D-01
normal_time_p_vs_t(74)=1.18327661643170D-01
normal_time_p_vs_t(75)=1.19786096256685D-01
normal_time_p_vs_t(76)=1.21341759844434D-01
normal_time_p_vs_t(77)=1.22897423432183D-01
normal_time_p_vs_t(78)=1.24355858045698D-01
normal_time_p_vs_t(79)=1.25911521633447D-01
normal_time_p_vs_t(80)=1.27467185221196D-01
normal_time_p_vs_t(81)=1.28925619834711D-01
normal_time_p_vs_t(82)=1.30481283422460D-01
normal_time_p_vs_t(83)=1.32036947010209D-01
normal_time_p_vs_t(84)=1.33495381623724D-01
normal_time_p_vs_t(85)=1.35051045211473D-01
normal_time_p_vs_t(86)=1.36509479824988D-01
normal_time_p_vs_t(87)=1.38065143412737D-01
normal_time_p_vs_t(88)=1.39620807000486D-01
normal_time_p_vs_t(89)=1.41079241614001D-01
normal_time_p_vs_t(90)=1.42634905201750D-01
normal_time_p_vs_t(91)=1.44093339815265D-01
normal_time_p_vs_t(92)=1.45649003403014D-01
normal_time_p_vs_t(93)=1.47204666990763D-01
normal_time_p_vs_t(94)=1.48760330578512D-01
normal_time_p_vs_t(95)=1.50218765192027D-01
normal_time_p_vs_t(96)=1.51774428779776D-01
normal_time_p_vs_t(97)=1.53330092367526D-01
normal_time_p_vs_t(98)=1.54788526981040D-01
normal_time_p_vs_t(99)=1.56344190568790D-01
normal_time_p_vs_t(100)=1.57802625182304D-01
normal_time_p_vs_t(101)=1.59358288770053D-01
```

```
normal_time_p_vs_t(102)=1.60913952357803D-01
normal_time_p_vs_t(103)=1.62372386971317D-01
normal_time_p_vs_t(104)=1.63928050559067D-01
normal_time_p_vs_t(105)=1.65483714146816D-01
normal_time_p_vs_t(106)=1.66942148760331D-01
normal_time_p_vs_t(107)=1.68497812348080D-01
normal_time_p_vs_t(108)=1.70053475935829D-01
normal_time_p_vs_t(109)=1.71511910549344D-01
normal_time_p_vs_t(110)=1.73067574137093D-01
normal_time_p_vs_t(111)=1.74526008750608D-01
normal_time_p_vs_t(112)=1.76081672338357D-01
normal_time_p_vs_t(113)=1.77637335926106D-01
normal_time_p_vs_t(114)=1.79095770539621D-01
normal_time_p_vs_t(115)=1.80651434127370D-01
normal_time_p_vs_t(116)=1.82207097715119D-01
normal_time_p_vs_t(117)=1.83665532328634D-01
normal_time_p_vs_t(118)=1.85221195916383D-01
normal_time_p_vs_t(119)=1.86776859504132D-01
normal_time_p_vs_t(120)=1.88235294117647D-01
normal_time_p_vs_t(121)=1.89790957705396D-01
normal_time_p_vs_t(122)=1.91249392318911D-01
normal_time_p_vs_t(123)=1.92805055906660D-01
normal_time_p_vs_t(124)=1.95819154107924D-01
normal_time_p_vs_t(125)=1.97374817695673D-01
normal_time_p_vs_t(126)=1.98930481283422D-01
normal_time_p_vs_t(127)=2.00388915896937D-01
normal_time_p_vs_t(128)=2.01944579484686D-01
normal_time_p_vs_t(129)=2.03500243072436D-01
normal_time_p_vs_t(130)=2.04861448711716D-01
normal_time_p_vs_t(131)=2.06514341273700D-01
normal_time_p_vs_t(132)=2.08070004861449D-01
normal_time_p_vs_t(133)=2.09528439474964D-01
normal_time_p_vs_t(134)=2.11084103062713D-01
normal_time_p_vs_t(135)=2.12542537676228D-01
normal_time_p_vs_t(136)=2.14098201263977D-01
normal_time_p_vs_t(137)=2.15653864851726D-01
normal_time_p_vs_t(138)=2.17112299465241D-01
normal_time_p_vs_t(139)=2.18570734078755D-01
normal_time_p_vs_t(140)=2.20223626640739D-01
normal_time_p_vs_t(141)=2.21682061254254D-01
normal_time_p_vs_t(142)=2.23237724842003D-01
normal_time_p_vs_t(143)=2.24793388429752D-01
normal_time_p_vs_t(144)=2.26251823043267D-01
normal_time_p_vs_t(145)=2.27807486631016D-01
normal_time_p_vs_t(146)=2.29265921244531D-01
normal_time_p_vs_t(147)=2.30821584832280D-01
normal_time_p_vs_t(148)=2.32280019445795D-01
normal_time_p_vs_t(149)=2.33835683033544D-01
```

```
normal_time_p_vs_t(150)=2.35391346621293D-01
normal_time_p_vs_t(151)=2.36947010209042D-01
normal_time_p_vs_t(152)=2.38405444822557D-01
normal_time_p_vs_t(153)=2.39961108410306D-01
normal_time_p_vs_t(154)=2.41516771998055D-01
normal_time_p_vs_t(155)=2.42975206611570D-01
normal_time_p_vs_t(156)=2.44530870199319D-01
normal_time_p_vs_t(157)=2.45989304812834D-01
normal_time_p_vs_t(158)=2.47544968400583D-01
normal_time_p_vs_t(159)=2.49100631988333D-01
normal_time_p_vs_t(160)=2.50559066601847D-01
normal_time_p_vs_t(161)=2.52114730189596D-01
normal_time_p_vs_t(162)=2.53670393777346D-01
normal_time_p_vs_t(163)=2.55128828390860D-01
normal_time_p_vs_t(164)=2.56684491978610D-01
normal_time_p_vs_t(165)=2.58240155566359D-01
normal_time_p_vs_t(166)=2.59698590179874D-01
normal_time_p_vs_t(167)=2.61254253767623D-01
normal_time_p_vs_t(168)=2.62809917355372D-01
normal_time_p_vs_t(169)=2.64268351968887D-01
normal_time_p_vs_t(170)=2.65824015556636D-01
normal_time_p_vs_t(171)=2.67379679144385D-01
normal_time_p_vs_t(172)=2.68838113757900D-01
normal_time_p_vs_t(173)=2.70393777345649D-01
normal_time_p_vs_t(174)=2.71949440933398D-01
normal_time_p_vs_t(175)=2.71852211959164D-01
normal_time_p_vs_t(176)=2.73407875546913D-01
normal_time_p_vs_t(177)=2.74963539134662D-01
normal_time_p_vs_t(178)=2.76421973748177D-01
normal_time_p_vs_t(179)=2.79533300923675D-01
normal_time_p_vs_t(180)=2.80991735537190D-01
normal_time_p_vs_t(181)=2.82547399124939D-01
normal_time_p_vs_t(182)=2.84103062712688D-01
normal_time_p_vs_t(183)=2.85561497326203D-01
normal_time_p_vs_t(184)=2.87117160913952D-01
normal_time_p_vs_t(185)=2.88672824501702D-01
normal_time_p_vs_t(186)=2.90131259115216D-01
normal_time_p_vs_t(187)=2.91686922702965D-01
normal_time_p_vs_t(188)=2.93242586290715D-01
normal_time_p_vs_t(189)=2.96256684491979D-01
normal_time_p_vs_t(190)=3.03840544482256D-01
normal_time_p_vs_t(191)=3.05396208070005D-01
normal_time_p_vs_t(192)=3.06757413709285D-01
normal_time_p_vs_t(193)=3.08410306271269D-01
normal_time_p_vs_t(194)=3.09965969859018D-01
normal_time_p_vs_t(195)=3.11424404472533D-01
normal_time_p_vs_t(196)=3.12980068060282D-01
normal_time_p_vs_t(197)=3.14535731648031D-01
```

```
normal_time_p_vs_t(198)=3.15994166261546D-01
normal_time_p_vs_t(199)=3.17549829849295D-01
normal_time_p_vs_t(200)=3.19105493437044D-01
normal_time_p_vs_t(201)=3.20466699076325D-01
normal_time_p_vs_t(202)=3.22119591638308D-01
normal_time_p_vs_t(203)=3.23675255226057D-01
normal_time_p_vs_t(204)=3.25133689839572D-01
normal_time_p_vs_t(205)=3.26689353427321D-01
normal_time_p_vs_t(206)=3.28245017015071D-01
normal_time_p_vs_t(207)=3.29703451628585D-01
normal_time_p_vs_t(208)=3.31259115216334D-01
normal_time_p_vs_t(209)=3.32814778804084D-01
normal_time_p_vs_t(210)=3.34175984443364D-01
normal_time_p_vs_t(211)=3.35828877005348D-01
normal_time_p_vs_t(212)=3.37384540593097D-01
normal_time_p_vs_t(213)=3.38842975206612D-01
normal_time_p_vs_t(214)=3.40398638794361D-01
normal_time_p_vs_t(215)=3.41954302382110D-01
normal_time_p_vs_t(216)=3.43412736995625D-01
normal_time_p_vs_t(217)=3.44968400583374D-01
normal_time_p_vs_t(218)=3.46524064171123D-01
normal_time_p_vs_t(219)=3.47885269810404D-01
normal_time_p_vs_t(220)=3.49538162372387D-01
normal_time_p_vs_t(221)=3.51093825960136D-01
normal_time_p_vs_t(222)=3.52552260573651D-01
normal_time_p_vs_t(223)=3.54107924161400D-01
normal_time_p_vs_t(224)=3.55663587749149D-01
normal_time_p_vs_t(225)=3.57122022362664D-01
normal_time_p_vs_t(226)=3.58677685950413D-01
normal_time_p_vs_t(227)=3.60233349538162D-01
normal_time_p_vs_t(228)=3.61594555177443D-01
normal_time_p_vs_t(229)=3.63247447739426D-01
normal_time_p_vs_t(230)=3.64803111327175D-01
normal_time_p_vs_t(231)=3.66261545940690D-01
normal_time_p_vs_t(232)=3.67817209528439D-01
normal_time_p_vs_t(233)=3.69372873116189D-01
normal_time_p_vs_t(234)=3.70831307729703D-01
normal_time_p_vs_t(235)=3.72386971317453D-01
normal_time_p_vs_t(236)=3.73942634905202D-01
normal_time_p_vs_t(237)=3.75303840544482D-01
normal_time_p_vs_t(238)=3.76956733106466D-01
normal_time_p_vs_t(239)=3.78512396694215D-01
normal_time_p_vs_t(240)=3.79970831307730D-01
normal_time_p_vs_t(241)=3.81526494895479D-01
normal_time_p_vs_t(242)=3.83082158483228D-01
normal_time_p_vs_t(243)=3.84540593096743D-01
normal_time_p_vs_t(244)=3.86096256684492D-01
normal_time_p_vs_t(245)=3.87651920272241D-01
```

```
normal_time_p_vs_t(246)=3.89110354885756D-01
normal_time_p_vs_t(247)=3.90666018473505D-01
normal_time_p_vs_t(248)=3.92221682061254D-01
normal_time_p_vs_t(249)=3.93680116674769D-01
normal_time_p_vs_t(250)=3.95138551288284D-01
normal_time_p_vs_t(251)=3.96791443850267D-01
normal_time_p_vs_t(252)=3.98249878463782D-01
normal_time_p_vs_t(253)=3.99805542051531D-01
normal_time_p_vs_t(254)=4.01361205639281D-01
normal_time_p_vs_t(255)=4.04375303840544D-01
normal_time_p_vs_t(256)=4.05930967428294D-01
normal_time_p_vs_t(257)=4.07389402041808D-01
normal_time_p_vs_t(258)=4.08847836655323D-01
normal_time_p_vs_t(259)=4.10500729217307D-01
normal_time_p_vs_t(260)=4.11959163830822D-01
normal_time_p_vs_t(261)=4.13514827418571D-01
normal_time_p_vs_t(262)=4.15070491006320D-01
normal_time_p_vs_t(263)=4.16528925619835D-01
normal_time_p_vs_t(264)=4.18084589207584D-01
normal_time_p_vs_t(265)=4.19640252795333D-01
normal_time_p_vs_t(266)=4.21098687408848D-01
normal_time_p_vs_t(267)=4.22557122022363D-01
normal_time_p_vs_t(268)=4.24210014584346D-01
normal_time_p_vs_t(269)=4.25765678172095D-01
normal_time_p_vs_t(270)=4.27224112785610D-01
normal_time_p_vs_t(271)=4.28779776373359D-01
normal_time_p_vs_t(272)=4.30335439961108D-01
normal_time_p_vs_t(273)=4.31793874574623D-01
normal_time_p_vs_t(274)=4.33349538162372D-01
normal_time_p_vs_t(275)=4.34905201750122D-01
normal_time_p_vs_t(276)=4.36266407389402D-01
normal_time_p_vs_t(277)=4.37919299951386D-01
normal_time_p_vs_t(278)=4.39474963539135D-01
normal_time_p_vs_t(279)=4.40933398152650D-01
normal_time_p_vs_t(280)=4.42489061740399D-01
normal_time_p_vs_t(281)=4.44044725328148D-01
normal_time_p_vs_t(282)=4.45503159941663D-01
normal_time_p_vs_t(283)=4.47058823529412D-01
normal_time_p_vs_t(284)=4.48614487117161D-01
normal_time_p_vs_t(285)=4.49975692756441D-01
normal_time_p_vs_t(286)=4.51628585318425D-01
normal_time_p_vs_t(287)=4.53184248906174D-01
normal_time_p_vs_t(288)=4.54642683519689D-01
normal_time_p_vs_t(289)=4.56198347107438D-01
normal_time_p_vs_t(290)=4.57754010695187D-01
normal_time_p_vs_t(291)=4.59212445308702D-01
normal_time_p_vs_t(292)=4.60768108896451D-01
normal_time_p_vs_t(293)=4.62323772484200D-01
```

```
normal_time_p_vs_t(294)=4.63684978123481D-01
normal_time_p_vs_t(295)=4.66893534273213D-01
normal_time_p_vs_t(296)=4.68351968886728D-01
normal_time_p_vs_t(297)=4.69907632474477D-01
normal_time_p_vs_t(298)=4.71463296062227D-01
normal_time_p_vs_t(299)=4.72921730675741D-01
normal_time_p_vs_t(300)=4.74477394263491D-01
normal_time_p_vs_t(301)=4.76033057851240D-01
normal_time_p_vs_t(302)=4.77394263490520D-01
normal_time_p_vs_t(303)=4.79047156052504D-01
normal_time_p_vs_t(304)=4.80602819640253D-01
normal_time_p_vs_t(305)=4.82061254253768D-01
normal_time_p_vs_t(306)=4.83616917841517D-01
normal_time_p_vs_t(307)=4.85172581429266D-01
normal_time_p_vs_t(308)=4.86631016042781D-01
normal_time_p_vs_t(309)=4.88186679630530D-01
normal_time_p_vs_t(310)=4.89742343218279D-01
normal_time_p_vs_t(311)=4.91298006806028D-01
normal_time_p_vs_t(312)=4.92756441419543D-01
normal_time_p_vs_t(313)=4.95867768595041D-01
normal_time_p_vs_t(314)=4.97228974234322D-01
normal_time_p_vs_t(315)=4.98881866796305D-01
normal_time_p_vs_t(316)=5.00437530384054D-01
normal_time_p_vs_t(317)=5.01895964997569D-01
normal_time_p_vs_t(318)=5.03451628585318D-01
normal_time_p_vs_t(319)=5.05007292173068D-01
normal_time_p_vs_t(320)=5.06465726786582D-01
normal_time_p_vs_t(321)=5.08021390374332D-01
normal_time_p_vs_t(322)=5.12591152163345D-01
normal_time_p_vs_t(323)=5.14146815751094D-01
normal_time_p_vs_t(324)=5.15605250364609D-01
normal_time_p_vs_t(325)=5.17160913952358D-01
normal_time_p_vs_t(326)=5.18716577540107D-01
normal_time_p_vs_t(327)=5.20175012153622D-01
normal_time_p_vs_t(328)=5.21730675741371D-01
normal_time_p_vs_t(329)=5.26300437530384D-01
normal_time_p_vs_t(330)=5.27856101118133D-01
normal_time_p_vs_t(331)=5.29314535731648D-01
normal_time_p_vs_t(332)=5.30870199319397D-01
normal_time_p_vs_t(333)=5.32425862907146D-01
normal_time_p_vs_t(334)=5.36995624696159D-01
normal_time_p_vs_t(335)=5.38356830335440D-01
normal_time_p_vs_t(336)=5.40009722897423D-01
normal_time_p_vs_t(337)=5.41565386485173D-01
normal_time_p_vs_t(338)=5.43023821098687D-01
normal_time_p_vs_t(339)=5.47593582887701D-01
normal_time_p_vs_t(340)=5.49149246475450D-01
normal_time_p_vs_t(341)=5.50704910063199D-01
```

```
normal_time_p_vs_t(342)=5.52066115702479D-01
normal_time_p_vs_t(343)=5.53719008264463D-01
normal_time_p_vs_t(344)=5.58288770053476D-01
normal_time_p_vs_t(345)=5.59844433641225D-01
normal_time_p_vs_t(346)=5.61302868254740D-01
normal_time_p_vs_t(347)=5.62858531842489D-01
normal_time_p_vs_t(348)=5.64414195430238D-01
normal_time_p_vs_t(349)=5.65775401069519D-01
normal_time_p_vs_t(350)=5.67428293631502D-01
normal_time_p_vs_t(351)=5.70442391832766D-01
normal_time_p_vs_t(352)=5.73553719008264D-01
normal_time_p_vs_t(353)=5.75012153621779D-01
normal_time_p_vs_t(354)=5.76567817209528D-01
normal_time_p_vs_t(355)=5.78123480797278D-01
normal_time_p_vs_t(356)=5.79484686436558D-01
normal_time_p_vs_t(357)=5.81137578998542D-01
normal_time_p_vs_t(358)=5.82693242586291D-01
normal_time_p_vs_t(359)=5.84151677199806D-01
normal_time_p_vs_t(360)=5.85707340787555D-01
normal_time_p_vs_t(361)=5.87263004375304D-01
normal_time_p_vs_t(362)=5.88818667963053D-01
normal_time_p_vs_t(363)=5.90277102576568D-01
normal_time_p_vs_t(364)=5.91832766164317D-01
normal_time_p_vs_t(365)=5.93388429752066D-01
normal_time_p_vs_t(366)=5.94846864365581D-01
normal_time_p_vs_t(367)=5.96402527953330D-01
normal_time_p_vs_t(368)=5.97958191541079D-01
normal_time_p_vs_t(369)=5.99416626154594D-01
normal_time_p_vs_t(370)=6.00972289742343D-01
normal_time_p_vs_t(371)=6.02527953330092D-01
normal_time_p_vs_t(372)=6.03986387943607D-01
normal_time_p_vs_t(373)=6.05542051531356D-01
normal_time_p_vs_t(374)=6.07097715119105D-01
normal_time_p_vs_t(375)=6.08556149732620D-01
normal_time_p_vs_t(376)=6.10111813320369D-01
normal_time_p_vs_t(377)=6.11667476908119D-01
normal_time_p_vs_t(378)=6.13125911521634D-01
normal_time_p_vs_t(379)=6.14681575109383D-01
normal_time_p_vs_t(380)=6.16237238697132D-01
normal_time_p_vs_t(381)=6.17695673310647D-01
normal_time_p_vs_t(382)=6.19251336898396D-01
normal_time_p_vs_t(383)=6.20807000486145D-01
normal_time_p_vs_t(384)=6.22265435099660D-01
normal_time_p_vs_t(385)=6.23821098687409D-01
normal_time_p_vs_t(386)=6.25376762275158D-01
normal_time_p_vs_t(387)=6.26835196888673D-01
normal_time_p_vs_t(388)=6.28390860476422D-01
normal_time_p_vs_t(389)=6.29946524064171D-01
```

```
normal_time_p_vs_t(390)=6.31404958677686D-01
normal_time_p_vs_t(391)=6.32960622265435D-01
normal_time_p_vs_t(392)=6.34516285853184D-01
normal_time_p_vs_t(393)=6.35974720466699D-01
normal_time_p_vs_t(394)=6.37530384054448D-01
normal_time_p_vs_t(395)=6.39086047642197D-01
normal_time_p_vs_t(396)=6.40544482255712D-01
normal_time_p_vs_t(397)=6.42100145843461D-01
normal_time_p_vs_t(398)=6.43655809431211D-01
normal_time_p_vs_t(399)=6.45114244044725D-01
normal_time_p_vs_t(400)=6.46669907632475D-01
normal_time_p_vs_t(401)=6.48225571220224D-01
normal_time_p_vs_t(402)=6.49684005833739D-01
normal_time_p_vs_t(403)=6.51239669421488D-01
normal_time_p_vs_t(404)=6.52795333009237D-01
normal_time_p_vs_t(405)=6.54253767622752D-01
normal_time_p_vs_t(406)=6.55809431210501D-01
normal_time_p_vs_t(407)=6.57365094798250D-01
normal_time_p_vs_t(408)=6.58823529411765D-01
normal_time_p_vs_t(409)=6.60379192999514D-01
normal_time_p_vs_t(410)=6.61934856587263D-01
normal_time_p_vs_t(411)=6.63393291200778D-01
normal_time_p_vs_t(412)=6.64948954788527D-01
normal_time_p_vs_t(413)=6.66504618376276D-01
normal_time_p_vs_t(414)=6.67963052989791D-01
normal_time_p_vs_t(415)=6.69518716577540D-01
normal_time_p_vs_t(416)=6.71074380165289D-01
normal_time_p_vs_t(417)=6.72532814778804D-01
normal_time_p_vs_t(418)=6.74088478366553D-01
normal_time_p_vs_t(419)=6.75644141954302D-01
normal_time_p_vs_t(420)=6.77102576567817D-01
normal_time_p_vs_t(421)=6.78658240155566D-01
normal_time_p_vs_t(422)=6.80213903743315D-01
normal_time_p_vs_t(423)=6.81575109382596D-01
normal_time_p_vs_t(424)=6.83228001944579D-01
normal_time_p_vs_t(425)=6.84783665532329D-01
normal_time_p_vs_t(426)=6.86242100145843D-01
normal_time_p_vs_t(427)=6.87797763733593D-01
normal_time_p_vs_t(428)=6.89353427321342D-01
normal_time_p_vs_t(429)=6.90811861934857D-01
normal_time_p_vs_t(430)=6.92367525522606D-01
normal_time_p_vs_t(431)=6.93923189110355D-01
normal_time_p_vs_t(432)=6.95284394749635D-01
normal_time_p_vs_t(433)=6.96937287311619D-01
normal_time_p_vs_t(434)=6.98492950899368D-01
normal_time_p_vs_t(435)=6.99951385512883D-01
normal_time_p_vs_t(436)=7.01507049100632D-01
normal_time_p_vs_t(437)=7.03062712688381D-01
```

```
normal_time_p_vs_t(438)=7.04521147301896D-01
normal_time_p_vs_t(439)=7.06076810889645D-01
normal_time_p_vs_t(440)=7.07632474477394D-01
normal_time_p_vs_t(441)=7.08993680116675D-01
normal_time_p_vs_t(442)=7.10646572678658D-01
normal_time_p_vs_t(443)=7.12202236266407D-01
normal_time_p_vs_t(444)=7.13660670879922D-01
normal_time_p_vs_t(445)=7.15216334467671D-01
normal_time_p_vs_t(446)=7.16771998055420D-01
normal_time_p_vs_t(447)=7.18230432668935D-01
normal_time_p_vs_t(448)=7.19786096256684D-01
normal_time_p_vs_t(449)=7.21341759844434D-01
normal_time_p_vs_t(450)=7.22702965483714D-01
normal_time_p_vs_t(451)=7.24355858045698D-01
normal_time_p_vs_t(452)=7.25911521633447D-01
normal_time_p_vs_t(453)=7.27369956246962D-01
normal_time_p_vs_t(454)=7.28925619834711D-01
normal_time_p_vs_t(455)=7.30481283422460D-01
normal_time_p_vs_t(456)=7.31939718035975D-01
normal_time_p_vs_t(457)=7.33495381623724D-01
normal_time_p_vs_t(458)=7.35051045211473D-01
normal_time_p_vs_t(459)=7.36412250850753D-01
normal_time_p_vs_t(460)=7.38065143412737D-01
normal_time_p_vs_t(461)=7.39620807000486D-01
normal_time_p_vs_t(462)=7.41079241614001D-01
normal_time_p_vs_t(463)=7.42634905201750D-01
normal_time_p_vs_t(464)=7.44190568789499D-01
normal_time_p_vs_t(465)=7.45649003403014D-01
normal_time_p_vs_t(466)=7.47204666990763D-01
normal_time_p_vs_t(467)=7.48760330578512D-01
normal_time_p_vs_t(468)=7.50121536217793D-01
normal_time_p_vs_t(469)=7.51774428779776D-01
normal_time_p_vs_t(470)=7.53330092367526D-01
normal_time_p_vs_t(471)=7.54788526981040D-01
normal_time_p_vs_t(472)=7.56344190568790D-01
normal_time_p_vs_t(473)=7.57899854156539D-01
normal_time_p_vs_t(474)=7.59358288770054D-01
normal_time_p_vs_t(475)=7.60913952357803D-01
normal_time_p_vs_t(476)=7.62469615945552D-01
normal_time_p_vs_t(477)=7.63830821584832D-01
normal_time_p_vs_t(478)=7.65483714146816D-01
normal_time_p_vs_t(479)=7.67039377734565D-01
normal_time_p_vs_t(480)=7.68497812348080D-01
normal_time_p_vs_t(481)=7.70053475935829D-01
normal_time_p_vs_t(482)=7.71609139523578D-01
normal_time_p_vs_t(483)=7.73067574137093D-01
normal_time_p_vs_t(484)=7.74623237724842D-01
normal_time_p_vs_t(485)=7.76178901312591D-01
```

```
normal_time_p_vs_t(486)=7.77540106951872D-01
normal_time_p_vs_t(487)=7.79192999513855D-01
normal_time_p_vs_t(488)=7.80748663101604D-01
normal_time_p_vs_t(489)=7.82207097715119D-01
normal_time_p_vs_t(490)=7.83762761302868D-01
normal_time_p_vs_t(491)=7.85318424890617D-01
normal_time_p_vs_t(492)=7.86776859504132D-01
normal_time_p_vs_t(493)=7.88332523091881D-01
normal_time_p_vs_t(494)=7.89888186679631D-01
normal_time_p_vs_t(495)=7.91249392318911D-01
normal_time_p_vs_t(496)=7.92902284880895D-01
normal_time_p_vs_t(497)=7.94457948468644D-01
normal_time_p_vs_t(498)=7.95916383082159D-01
normal_time_p_vs_t(499)=7.97472046669908D-01
normal_time_p_vs_t(500)=7.99027710257657D-01
normal_time_p_vs_t(501)=8.00486144871172D-01
normal_time_p_vs_t(502)=8.02041808458921D-01
normal_time_p_vs_t(503)=8.03597472046670D-01
normal_time_p_vs_t(504)=8.04958677685950D-01
normal_time_p_vs_t(505)=8.06611570247934D-01
normal_time_p_vs_t(506)=8.08167233835683D-01
normal_time_p_vs_t(507)=8.09625668449198D-01
normal_time_p_vs_t(508)=8.11181332036947D-01
normal_time_p_vs_t(509)=8.12736995624696D-01
normal_time_p_vs_t(510)=8.14195430238211D-01
normal_time_p_vs_t(511)=8.15751093825960D-01
normal_time_p_vs_t(512)=8.17306757413709D-01
normal_time_p_vs_t(513)=8.18667963052990D-01
normal_time_p_vs_t(514)=8.20320855614973D-01
normal_time_p_vs_t(515)=8.21876519202723D-01
normal_time_p_vs_t(516)=8.23334953816237D-01
normal_time_p_vs_t(517)=8.24890617403987D-01
normal_time_p_vs_t(518)=8.26446280991736D-01
normal_time_p_vs_t(519)=8.27904715605250D-01
normal_time_p_vs_t(520)=8.29460379193000D-01
normal_time_p_vs_t(521)=8.31016042780749D-01
normal_time_p_vs_t(522)=8.32377248420029D-01
normal_time_p_vs_t(523)=8.34030140982013D-01
normal_time_p_vs_t(524)=8.35585804569762D-01
normal_time_p_vs_t(525)=8.37044239183277D-01
normal_time_p_vs_t(526)=8.38599902771026D-01
normal_time_p_vs_t(527)=8.40155566358775D-01
normal_time_p_vs_t(528)=8.41614000972290D-01
normal_time_p_vs_t(529)=8.43169664560039D-01
normal_time_p_vs_t(530)=8.44725328147788D-01
normal_time_p_vs_t(531)=8.46086533787069D-01
normal_time_p_vs_t(532)=8.47739426349052D-01
normal_time_p_vs_t(533)=8.49295089936801D-01
```

```
normal_time_p_vs_t(534)=8.50753524550316D-01
normal_time_p_vs_t(535)=8.52309188138065D-01
normal_time_p_vs_t(536)=8.53864851725814D-01
normal_time_p_vs_t(537)=8.55323286339329D-01
normal_time_p_vs_t(538)=8.56878949927078D-01
normal_time_p_vs_t(539)=8.58434613514827D-01
normal_time_p_vs_t(540)=8.59795819154108D-01
normal_time_p_vs_t(541)=8.61448711716091D-01
normal_time_p_vs_t(542)=8.63004375303841D-01
normal_time_p_vs_t(543)=8.64462809917355D-01
normal_time_p_vs_t(544)=8.66018473505105D-01
normal_time_p_vs_t(545)=8.67379679144385D-01
normal_time_p_vs_t(546)=8.69032571706369D-01
normal_time_p_vs_t(547)=8.70588235294118D-01
normal_time_p_vs_t(548)=8.72046669907633D-01
normal_time_p_vs_t(549)=8.73505104521147D-01
normal_time_p_vs_t(550)=8.75157997083131D-01
normal_time_p_vs_t(551)=8.76616431696646D-01
normal_time_p_vs_t(552)=8.78172095284395D-01
normal_time_p_vs_t(553)=8.79727758872144D-01
normal_time_p_vs_t(554)=8.81088964511424D-01
normal_time_p_vs_t(555)=8.82741857073408D-01
normal_time_p_vs_t(556)=8.84297520661157D-01
normal_time_p_vs_t(557)=8.85755955274672D-01
normal_time_p_vs_t(558)=8.87214389888187D-01
normal_time_p_vs_t(559)=8.88867282450170D-01
normal_time_p_vs_t(560)=8.90325717063685D-01
normal_time_p_vs_t(561)=8.91881380651434D-01
normal_time_p_vs_t(562)=8.93437044239183D-01
normal_time_p_vs_t(563)=8.94798249878464D-01
normal_time_p_vs_t(564)=8.96451142440447D-01
normal_time_p_vs_t(565)=8.98006806028196D-01
normal_time_p_vs_t(566)=8.99465240641711D-01
normal_time_p_vs_t(567)=9.00923675255226D-01
normal_time_p_vs_t(568)=9.02576567817210D-01
normal_time_p_vs_t(569)=9.04035002430724D-01
normal_time_p_vs_t(570)=9.05590666018473D-01
normal_time_p_vs_t(571)=9.07146329606223D-01
normal_time_p_vs_t(572)=9.08507535245503D-01
normal_time_p_vs_t(573)=9.10160427807487D-01
normal_time_p_vs_t(574)=9.11716091395236D-01
normal_time_p_vs_t(575)=9.13174526008751D-01
normal_time_p_vs_t(576)=9.14632960622265D-01
normal_time_p_vs_t(577)=9.16285853184249D-01
normal_time_p_vs_t(578)=9.17744287797764D-01
normal_time_p_vs_t(579)=9.19299951385513D-01
normal_time_p_vs_t(580)=9.20855614973262D-01
normal_time_p_vs_t(581)=9.22216820612543D-01
```

```
normal_time_p_vs_t(582)=9.23869713174526D-01
normal_time_p_vs_t(583)=9.25425376762275D-01
normal_time_p_vs_t(584)=9.26883811375790D-01
normal_time_p_vs_t(585)=9.28342245989305D-01
normal_time_p_vs_t(586)=9.29995138551288D-01
normal_time_p_vs_t(587)=9.31453573164803D-01
normal_time_p_vs_t(588)=9.33009236752552D-01
normal_time_p_vs_t(589)=9.34564900340301D-01
normal_time_p_vs_t(590)=9.36023334953816D-01
normal_time_p_vs_t(591)=9.37578998541565D-01
normal_time_p_vs_t(592)=9.39134662129315D-01
normal_time_p_vs_t(593)=9.40593096742829D-01
normal_time_p_vs_t(594)=9.42051531356344D-01
normal_time_p_vs_t(595)=9.43704423918328D-01
normal_time_p_vs_t(596)=9.45162858531842D-01
normal_time_p_vs_t(597)=9.46718522119592D-01
normal_time_p_vs_t(598)=9.48176956733106D-01
normal_time_p_vs_t(599)=9.49732620320856D-01
normal_time_p_vs_t(600)=9.51288283908605D-01
normal_time_p_vs_t(601)=9.52843947496354D-01
normal_time_p_vs_t(602)=9.54302382109869D-01
normal_time_p_vs_t(603)=9.55760816723384D-01
normal_time_p_vs_t(604)=9.57413709285367D-01
normal_time_p_vs_t(605)=9.58872143898882D-01
normal_time_p_vs_t(606)=9.60427807486631D-01
normal_time_p_vs_t(607)=9.61886242100146D-01
normal_time_p_vs_t(608)=9.63441905687895D-01
normal_time_p_vs_t(609)=9.64997569275644D-01
normal_time_p_vs_t(610)=9.66553232863393D-01
normal_time_p_vs_t(611)=9.68011667476908D-01
normal_time_p_vs_t(612)=9.69470102090423D-01
normal_time_p_vs_t(613)=9.71122994652406D-01
normal_time_p_vs_t(614)=9.72581429265921D-01
normal_time_p_vs_t(615)=9.74137092853670D-01
normal_time_p_vs_t(616)=9.75692756441420D-01
normal_time_p_vs_t(617)=9.77151191054934D-01
normal_time_p_vs_t(618)=9.78706854642683D-01
normal_time_p_vs_t(619)=9.80262518230433D-01
normal_time_p_vs_t(620)=9.81720952843948D-01
normal_time_p_vs_t(621)=9.83276616431697D-01
normal_time_p_vs_t(622)=9.84832280019446D-01
normal_time_p_vs_t(623)=9.86290714632961D-01
normal_time_p_vs_t(624)=9.87846378220710D-01
normal_time_p_vs_t(625)=9.89402041808459D-01
normal_time_p_vs_t(626)=9.90860476421974D-01
normal_time_p_vs_t(627)=9.92416140009723D-01
normal_time_p_vs_t(628)=9.93971803597472D-01
normal_time_p_vs_t(629)=9.95430238210987D-01
```

```
normal_time_p_vs_t(630)=9.96985901798736D-01
normal_time_p_vs_t(631)=9.98541565386485D-01
normal_time_p_vs_t(632)=1.00000000000000D+00
!-----------------------------------------------



END SUBROUTINE PROFILE
!---------------------------------------------------------------------------------------------------------------------------------!
```

Appendix B
Input File

```
2                  ! CASE_FLAG               (integer) ! 1=STATIC, 2=DYNAMIC
8                  ! N_layers                (integer) ! change thetak(i), hk(i), materialprop(i) to match !!!
0.0035052D0        ! hk(1)                   (meters)
0.0015875D0        ! hk(2)                   (meters)
0.001058333D0      ! hk(3)                   (meters)
0.0015875D0        ! hk(4)                   (meters)
0.001058333D0      ! hk(5)                   (meters)
0.0015875D0        ! hk(6)                   (meters)
0.001058333D0      ! hk(7)                   (meters)
0.0015875D0        ! hk(8)                   (meters)
0.D0               ! thetak(1)=              (degrees) ! 0 degrees is parallel with cylinder axis, 90 is perpendicular
59.D0              ! thetak(2)=              (degrees)
0.D0               ! thetak(3)=              (degrees)
-59.D0             ! thetak(4)=              (degrees)
0.D0               ! thetak(5)=              (degrees)
59.D0              ! thetak(6)=              (degrees)
0.D0               ! thetak(7)=              (degrees)
-59.D0             ! thetak(8)=              (degrees)
4                  ! materialprop(1)=        (integer) ! 2=T300_5280
2                  ! materialprop(2)=        (integer)
6                  ! materialprop(3)=        (integer)
2                  ! materialprop(4)=        (integer)
6                  ! materialprop(5)=        (integer)
2                  ! materialprop(6)=        (integer)
6                  ! materialprop(7)=        (integer)
2                  ! materialprop(8)=        (integer)
3                  ! r_elements_per_layer(1) (integer) ! r_elements calculated from SUM(r_elements_per_layer)
4                  ! r_elements_per_layer(2) (integer)
4                  ! r_elements_per_layer(3) (integer)
4                  ! r_elements_per_layer(4) (integer)
4                  ! r_elements_per_layer(5) (integer)
4                  ! r_elements_per_layer(6) (integer)
4                  ! r_elements_per_layer(7) (integer)
4                  ! r_elements_per_layer(8) (integer)
0.0028448D0        ! r_layer_loc(0)          (meters)  ! Inside radius, 127 mm = 5", 254 mm = 10"
0.508D0            ! x_length                (meters)
20                 ! x_elements              (integer)
10000              ! n_timesteps             (integer)
0.0015D0           ! time                    (seconds)
413685437.5901D0   ! P_constant (P_max)      (Pa)      ! 10 kPa = 1.45 psi, 1 psi = 6894.76 Pa
914.4D0            ! V_max                   (m/s)     ! 1000 ft/s = 304.8 m/s

! 1=GRAPHITE_POLYMER, 2=T300_5280, 3=STAINLESS_STEEL_ENG, 4=STAINLESS_STEEL_SI, 5=FRONK_GRAPHITE_POLYMER
! 6=VISCO_ELASTIC

!  P_max=413685437.5901d0 ! Pa  (60,000 psi)       68947.572931684D0
!  V_max=835.81134336d0   ! m/s (2742.1632 ft/s)
```

# Appendix C
Raw Profile Data, Pressure vs. Position

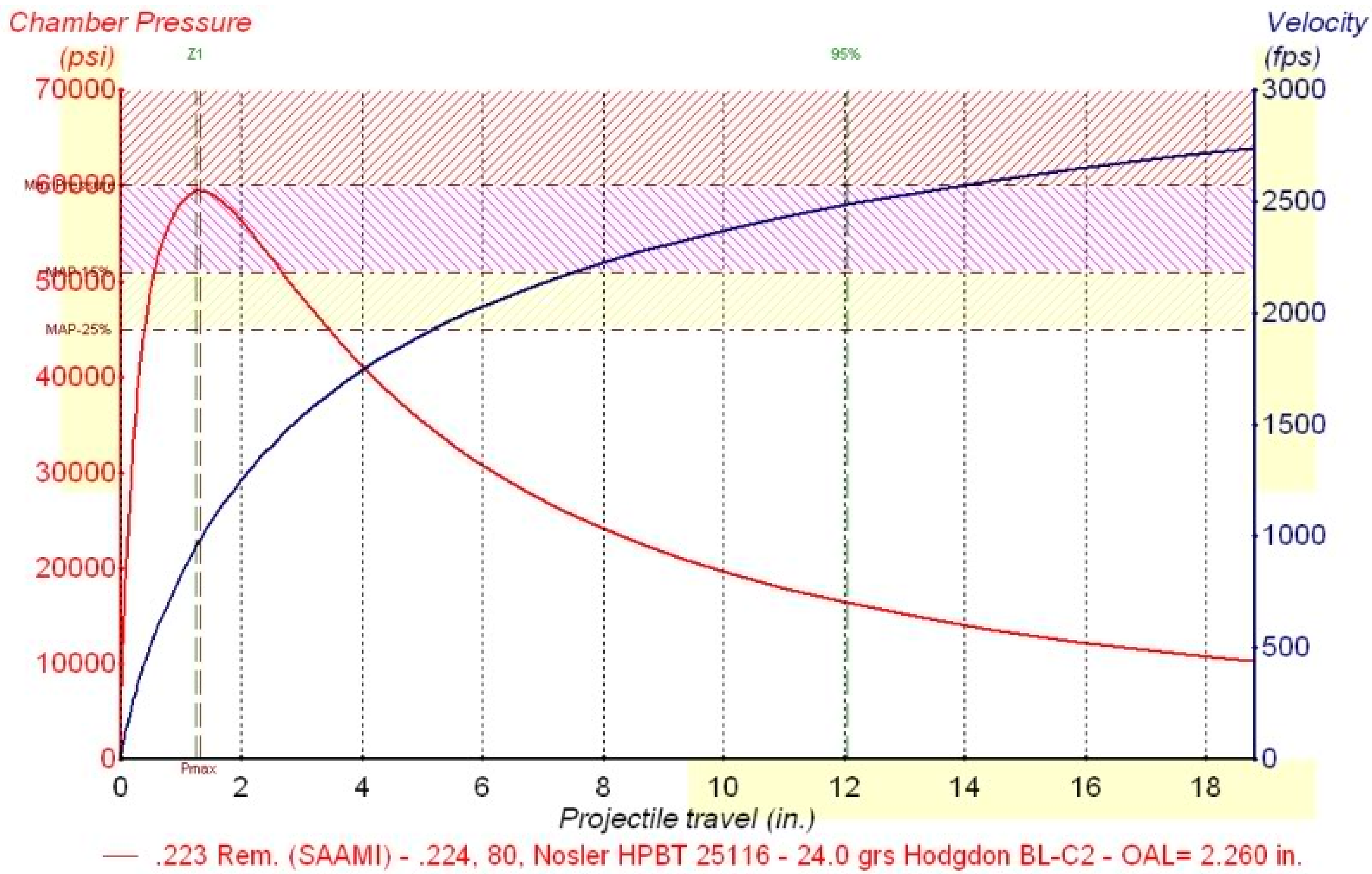



Figure C.1 Projectile Travel (in.)

Appendix D
Raw Profile Data, Pressure vs. Time

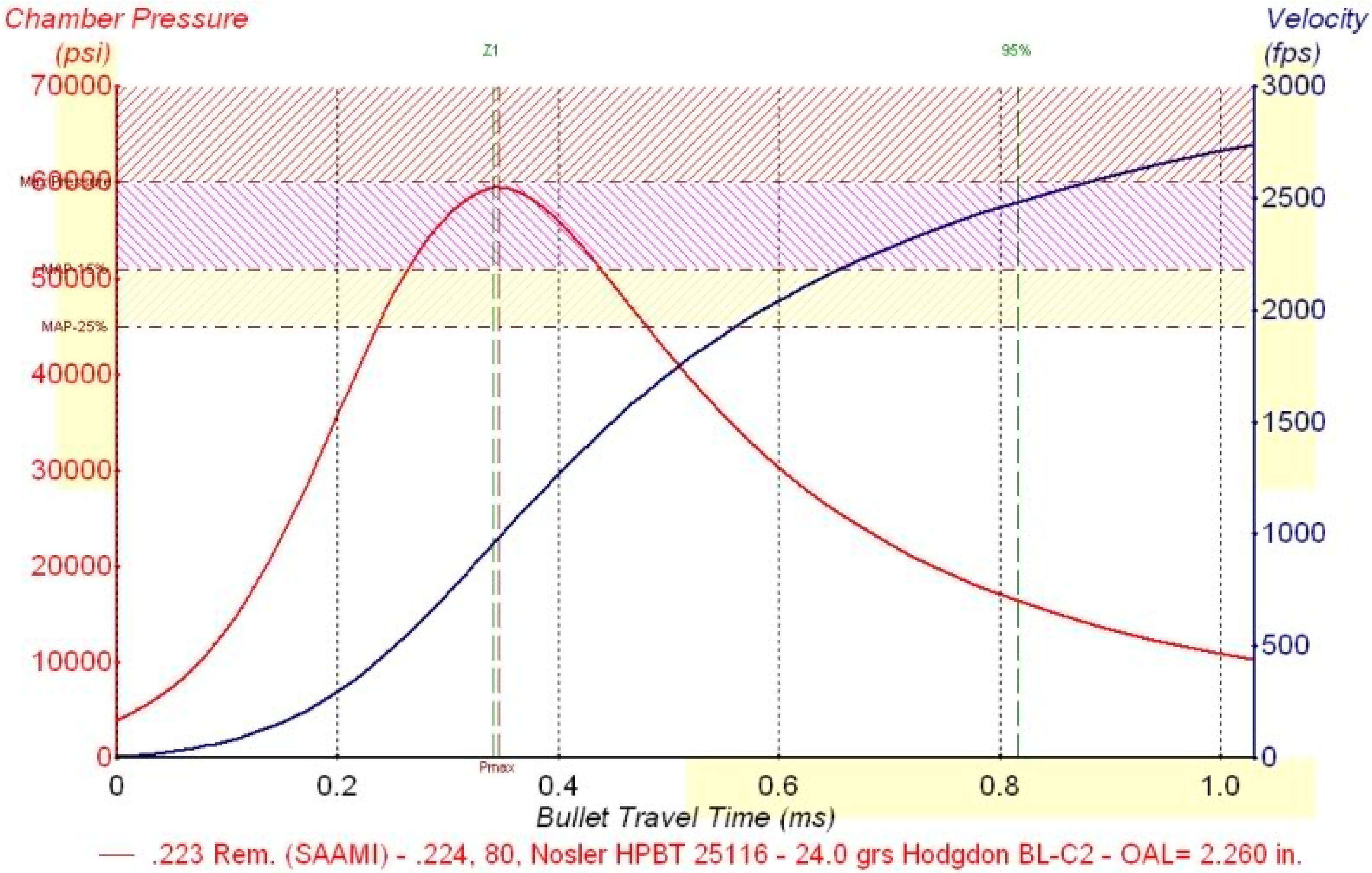



Figure D.1 Bullet Travel Time (ms)

# Appendix E
## Nodal Pressure Application Detail

Figure E.1 through Figure E.7 represent how the boundary conditions are interpolated and applied at each nodal location throughout time.

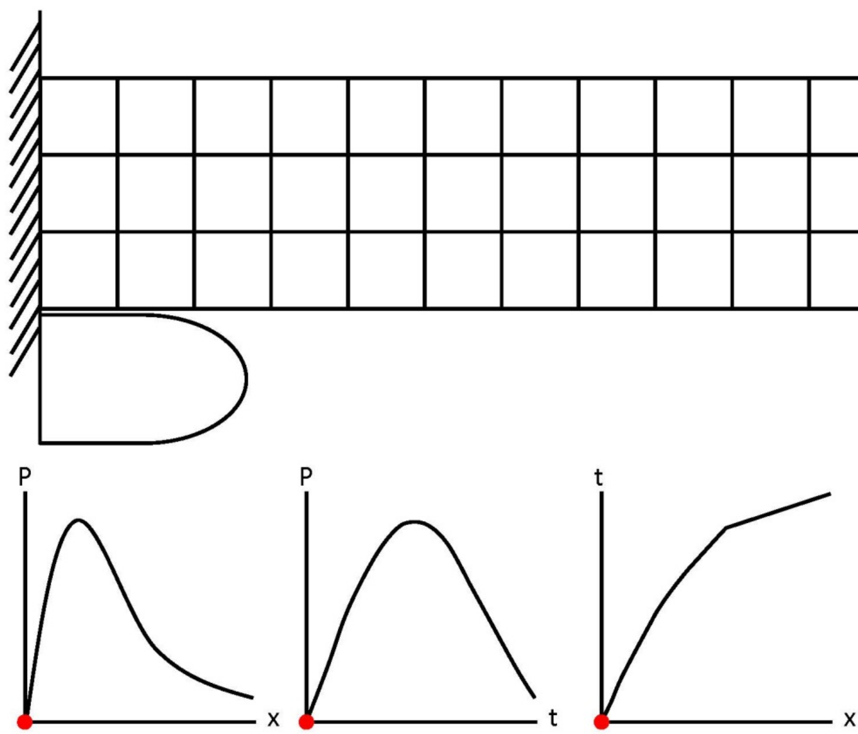



Figure E.1 Boundary Conditions, $t_0$

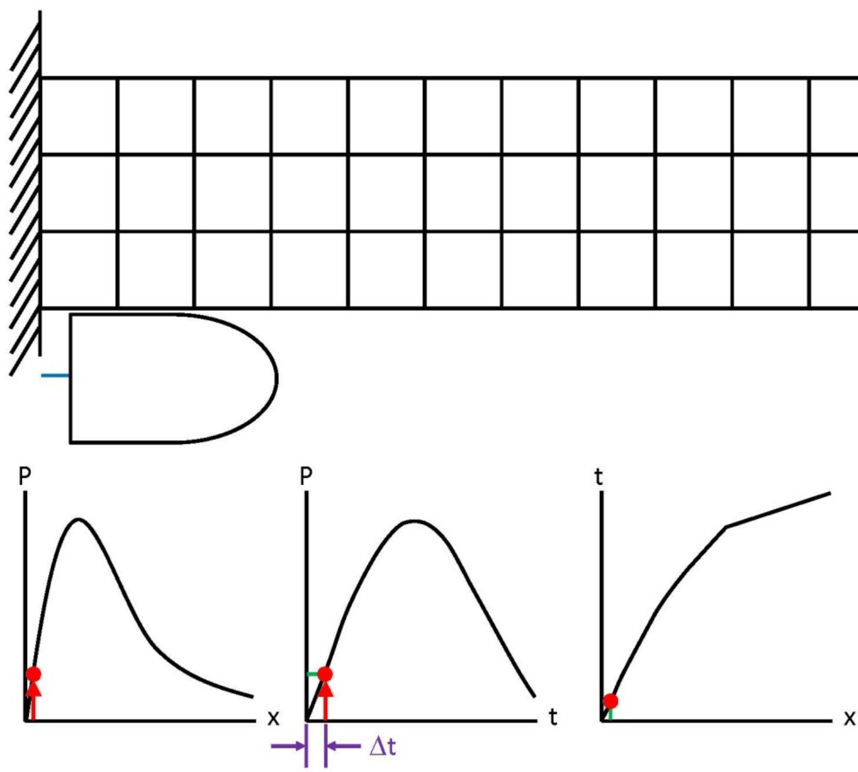



Figure E.2 Boundary Conditions, $t_1$

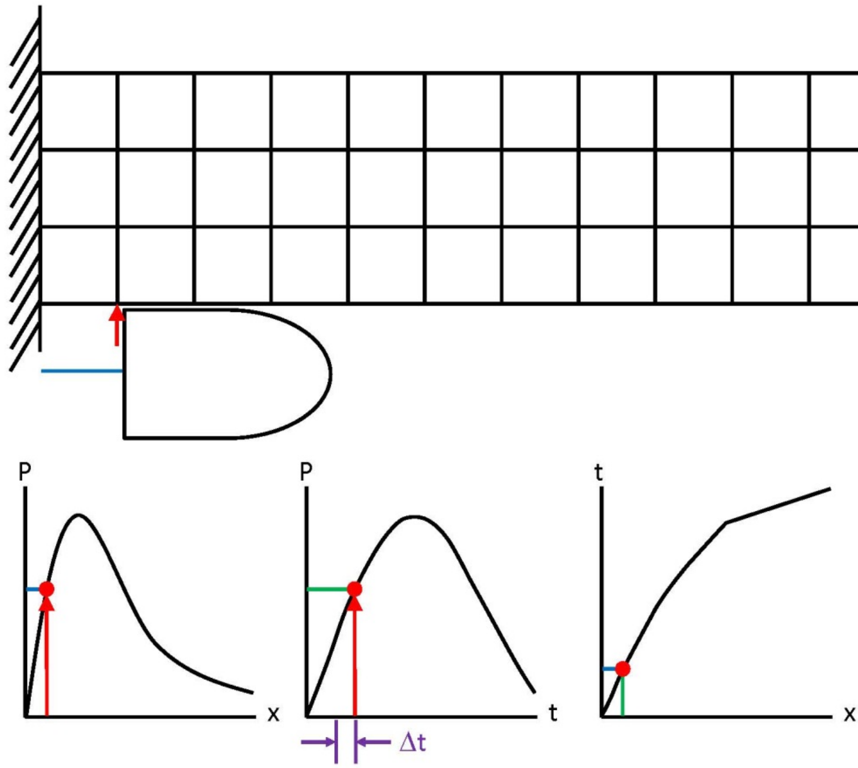



Figure E.3 Boundary Conditions, $t_2$

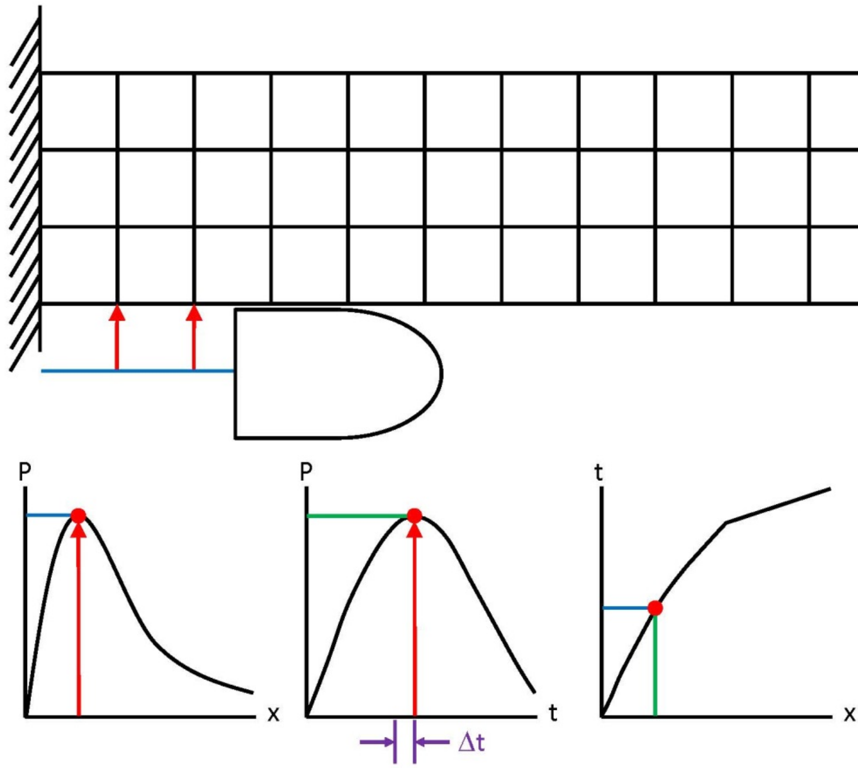



Figure E.4 Boundary Conditions, $t_3$

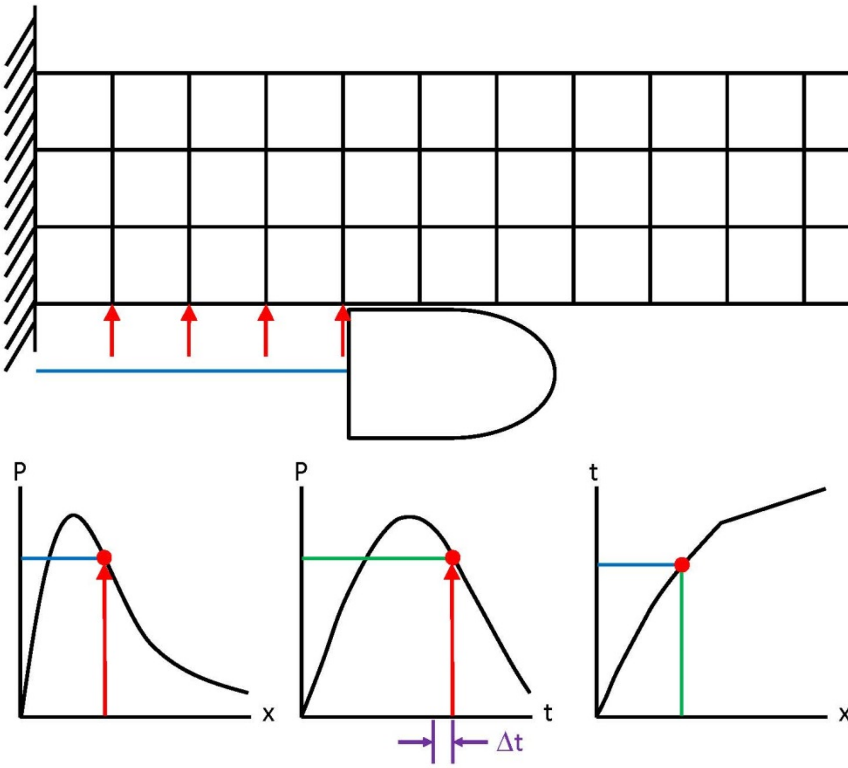



Figure E.5 Boundary Conditions, $t_4$

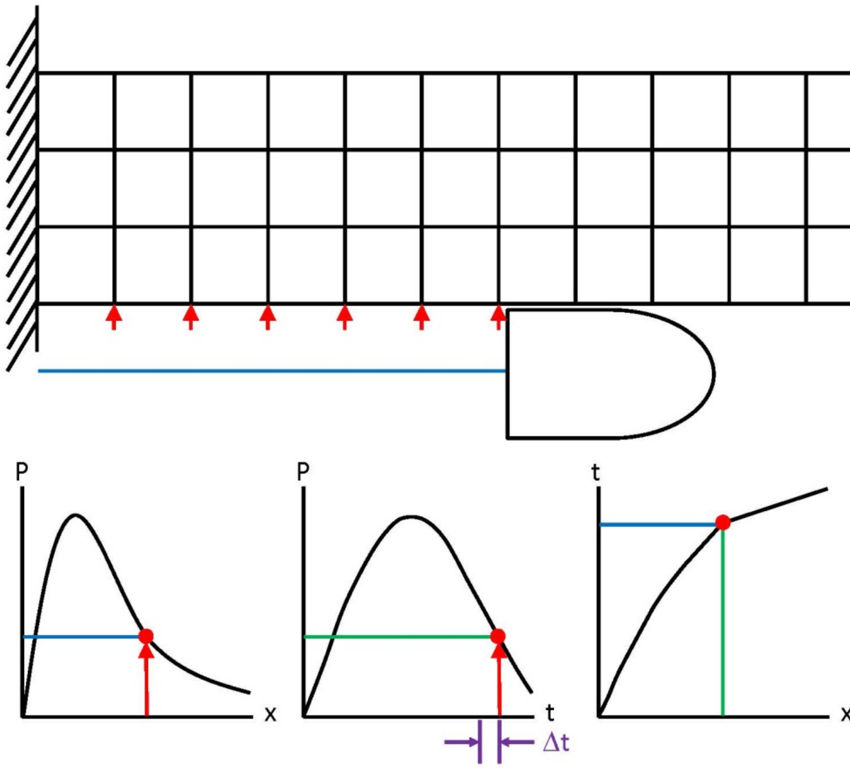



Figure E.6 Boundary Conditions, $t_5$

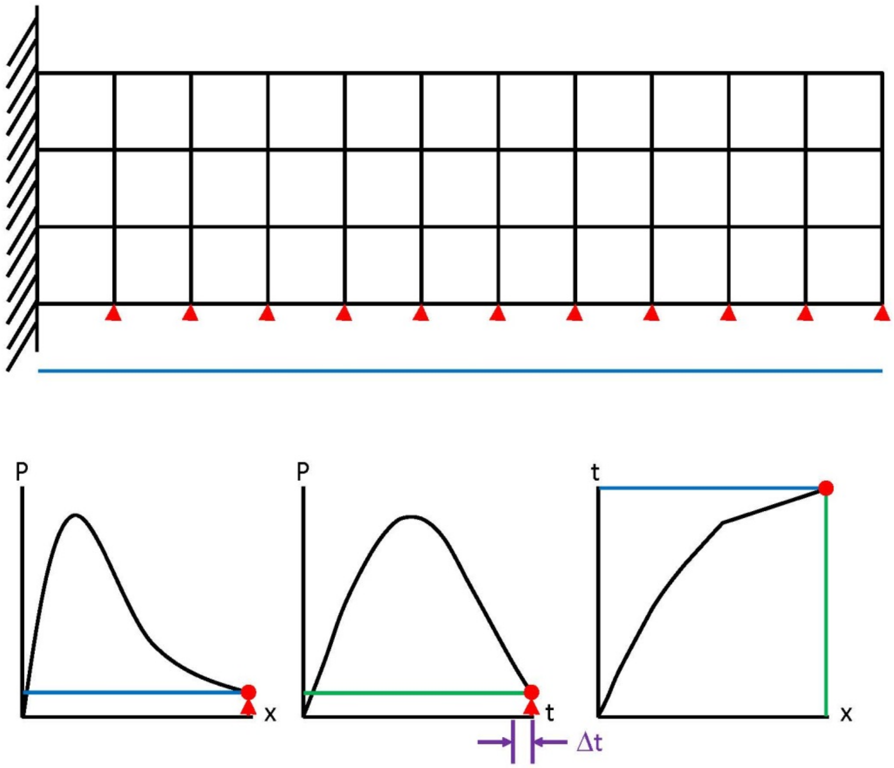



Figure E.7 Boundary Conditions, $t_6$